Cisco H.323 Signaling Interface User Guide

Cisco HSI Release 4.1
November 2007

Corporate Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 526-4100

Text Part Number: OL-4806-01 Rev. A14

Cisco H.323 Signaling Interface User Guide

CONTENTS

Preface

This preface describes the objectives, audience, organization, and conventions of the *Cisco H.323 Signaling Interface User Guide*, and explains how to find additional information on related products and services. It contains the following sections:

Document Objectives

This guide contains installation, configuration, system management, troubleshooting, and Man-Machine Language (MML) command information for the Cisco H.323 Signaling Interface (HSI).

This version of the *Cisco H.323 Signaling Interface User Guide* documents the Cisco H.323 Signaling Interface (HSI) software, Release 4.1. Cisco HSI, Release 4.1 is associated with the Cisco Media Gateway Controller Software, Release 9.4(1).

Audience

The intended audience is the system administrator, the system operator, and the system technician.

System Administrator

The system administrator is required to manage the host administrative functions, including:

- Configuring and maintaining system parameters
- Granting group and user IDs
- Managing all Cisco Public Switched Telephone Network (PSTN) Gateway (PGW 2200) files and directories

The system administrator should have an in-depth knowledge of UNIX and a basic knowledge of data and telecommunications networking.

System Operator

The system operator is assumed to have knowledge of the following:

- Telecommunications protocols
- Basic computer software operations
- Computer terminology and concepts
- Hierarchical file systems
- Common UNIX shell commands

System Technician

The system technician is assumed to have knowledge of the following:

- Telecommunications protocols
- Basic computer software operations
- Computer terminology and concepts
- Hierarchical file systems
- Common UNIX shell commands
- Log files
- Configuration of telephony switching systems
- Use of electrical and electronic telephony test equipment
- Basic troubleshooting techniques

Document Organization

This document is organized as follows:

- Preface
- Chapter 1, “Cisco H.323 Signaling Interface Overview.”
- Chapter 2, “Installing and Configuring Cisco HSI Software”
- Chapter 3, “Provisioning the Cisco HSI”
- Chapter 4, “Managing the Cisco HSI”
- Chapter 5, “Troubleshooting Cisco HSI Alarms”
- Appendix A, “MML User Interface and Command Reference”
- Appendix B, “Skeleton Configuration File”
- Appendix C, “Example HSI Configuration File”
- Appendix D, “E-ISUP Name-to-Cause Value Lookup”

- Appendix E, “E-ISUP Cause Value-to-Name Lookup”
- Appendix F, “H.323 Name-to-Cause Value Lookup”
- Appendix G, “H.323 Cause Value-to-Name Lookup”

Document Conventions

This manual uses the document conventions listed in this section.

Table 1 Document Conventions

Convention	Meaning	Comments and Examples
Boldface	Commands and keywords you enter literally as shown	**prov-sta**
Italics	Variables for which you supply values	**command** *interface type* You replace the variable with the type of interface.
`Courier`	Font used for screen displays, prompts, and scripts.	`Are you ready to continue? [Y]`
`Courier bold`	Font used to indicate what the user enters in examples of command environments.	Login: **`root`** Password: **`<password>`**
Square brackets ([ ])	Optional elements	**command** [abc] abc is optional (not required), but you can choose it.
Vertical bars (|)	Separated alternative elements	**command** [abc | def] You can choose either abc or def, or neither, but not both.
Braces ({ })	Required choices	**command** {abc | def} You must use either abc or def, but not both.
Braces with vertical bars within square brackets ([{ | }])	A required choice within an optional element	**command** [abc{ def | ghi}] You have three options: • No entry • abc def • abc ghi
A string	A nonquoted set of characters	For example, when setting an SNMP community string to public, do not use quotation marks around the string; otherwise, the string will include the quotation marks.

Table 1 Document Conventions (continued)

Convention	Meaning	Comments and Examples
System prompt	Denotes interactive sessions; indicates that the user enters commands at the prompt	The system prompt indicates the current command mode. For example, the prompt `Router (config)#` indicates global configuration mode.
Exclamation point (!) at the beginning of a line	A comment line	Comments are sometimes displayed.

Note Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the manual.

Tip Means *the following information will help you solve a problem*. The tip information might not be troubleshooting or even an action, but could be useful information, similar to a Timesaver.

Caution Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Related Documentation

The following sections provide the titles of documents related to the *Cisco H.323 Signaling Interface User Guide*.

Release Notes

For information regarding subsequent releases of the Cisco H.323 signaling interface, refer to:

- *Release Notes for Cisco H.323 Signaling Interface Release 4.1 and Related Patches*

Hardware Documentation

- *Cisco Media Gateway Controller Hardware Installation Guide*
- *Regulatory Compliance and Safety Information for Cisco Media Gateway Controller*
- *Cisco Media Gateway Hardware Installation Guide*

Software Documentation

- *Cisco Media Gateway Controller Software Release 9 Installation and Configuration Guide*
- *Cisco Media Gateway Controller Software Release 9 Provisioning Guide*
- *Cisco Media Gateway Controller Software Release 9 MML Command Reference Guide*
- *Cisco Media Gateway Controller Software Release 9 Messages Reference Guide*
- *Cisco Media Gateway Controller Software Release 9 Billing Interface Guide*
- *Cisco Media Gateway Controller Software Release 9 Operations, Maintenance, and Troubleshooting Guide*
- *Cisco Media Gateway Controller Software Release 9 Management Information Base Guide*
- *Cisco Media Gateway Controller Node Manager User's Guide 2.0*
- *Cisco Signaling Link Terminal*
- *Cisco Media Gateway Controller Online Documentation Notice*
- *Cisco Media Gateway Controller SLT Documentation Notice*

Related Documentation

- ITU Recommendation H.323, 2000
- ITU Recommendation H.225, 2001
- ITU Recommendation H.245, 2000
- ITU Recommendation H.246 Annex C

Obtaining Documentation

These sections explain how to obtain documentation from Cisco Systems.

World Wide Web

You can access the most current Cisco documentation on the World Wide Web at this URL:

http://www.cisco.com

Translated documentation is available at this URL:

http://www.cisco.com/public/countries_languages.shtml

Documentation CD-ROM

Cisco documentation and additional literature are available in a Cisco Documentation CD-ROM package, which is shipped with your product. The Documentation CD-ROM is updated monthly and may be more current than printed documentation. The CD-ROM package is available as a single unit or through an annual subscription.

Ordering Documentation

You can order Cisco documentation in these ways:

- Registered Cisco.com users (Cisco direct customers) can order Cisco product documentation from the Networking Products MarketPlace:

  http://www.cisco.com/cgi-bin/order/order_root.pl

- Registered Cisco.com users can order the Documentation CD-ROM through the online Subscription Store:

  http://www.cisco.com/go/subscription

- Nonregistered Cisco.com users can order documentation through a local account representative by calling Cisco Systems Corporate Headquarters (California, U.S.A.) at 408 526-7208 or, elsewhere in North America, by calling 800 553-NETS (6387).

Documentation Feedback

You can submit comments electronically on Cisco.com. In the Cisco Documentation home page, click the **Fax** or **Email** option in the "Leave Feedback" section at the bottom of the page.

You can e-mail your comments to bug-doc@cisco.com.

You can submit your comments by mail by writing to the following address:

Cisco Systems
Attn: Document Resource Connection
170 West Tasman Drive
San Jose, CA 95134-9883

We appreciate your comments.

Obtaining Technical Assistance

Cisco provides Cisco.com as a starting point for all technical assistance. Customers and partners can obtain online documentation, troubleshooting tips, and sample configurations from online tools by using the Cisco Technical Assistance Center (TAC) Web Site. Cisco.com registered users have complete access to the technical support resources on the Cisco TAC Web Site.

Cisco.com

Cisco.com is the foundation of a suite of interactive, networked services that provides immediate, open access to Cisco information, networking solutions, services, programs, and resources at any time, from anywhere in the world.

Cisco.com is a highly integrated Internet application and a powerful, easy-to-use tool that provides a broad range of features and services to help you with these tasks:

- Streamline business processes and improve productivity
- Resolve technical issues with online support
- Download and test software packages
- Order Cisco learning materials and merchandise

- Register for online skill assessment, training, and certification programs

If you want to obtain customized information and service, you can self-register on Cisco.com. To access Cisco.com, go to this URL:

http://www.cisco.com

Technical Assistance Center

The Cisco Technical Assistance Center (TAC) is available to all customers who need technical assistance with a Cisco product, technology, or solution. Two levels of support are available: the Cisco TAC Web Site and the Cisco TAC Escalation Center.

Cisco TAC inquiries are categorized according to the urgency of the issue:

- Priority level 4 (P4)—You need information or assistance concerning Cisco product capabilities, product installation, or basic product configuration.
- Priority level 3 (P3)—Your network performance is degraded. Network functionality is noticeably impaired, but most business operations continue.
- Priority level 2 (P2)—Your production network is severely degraded, affecting significant aspects of business operations. No workaround is available.
- Priority level 1 (P1)—Your production network is down, and a critical impact to business operations will occur if service is not restored quickly. No workaround is available.

The Cisco TAC resource that you choose is based on the priority of the problem and the conditions of service contracts, when applicable.

Cisco TAC Web Site

You can use the Cisco TAC Web Site to resolve P3 and P4 issues yourself, saving both cost and time. The site provides around-the-clock access to online tools, knowledge bases, and software. To access the Cisco TAC Web Site, go to this URL:

http://www.cisco.com/tac

All customers, partners, and resellers who have a valid Cisco service contract have complete access to the technical support resources on the Cisco TAC Web Site. The Cisco TAC Web Site requires a Cisco.com login ID and password. If you have a valid service contract but do not have a login ID or password, go to this URL to register:

http://www.cisco.com/register/

If you are a Cisco.com registered user, and you cannot resolve your technical issues by using the Cisco TAC Web Site, you can open a case online by using the TAC Case Open tool at this URL:

http://www.cisco.com/tac/caseopen

If you have Internet access, we recommend that you open P3 and P4 cases through the Cisco TAC Web Site.

Cisco TAC Escalation Center

The Cisco TAC Escalation Center addresses priority level 1 or priority level 2 issues. These classifications are assigned when severe network degradation significantly impacts business operations. When you contact the TAC Escalation Center with a P1 or P2 problem, a Cisco TAC engineer automatically opens a case.

To obtain a directory of toll-free Cisco TAC telephone numbers for your country, go to this URL:

http://www.cisco.com/warp/public/687/Directory/DirTAC.shtml

Before calling, please check with your network operations center to determine the level of Cisco support services to which your company is entitled: for example, SMARTnet, SMARTnet Onsite, or Network Supported Accounts (NSA). When you call the center, please have available your service agreement number and your product serial number.

Cisco H.323 Signaling Interface Overview

Introduction

This chapter provides an overview of the Cisco H.323 Signaling Interface (HSI) system and subsystems and contains the following sections:

Cisco HSI Overview

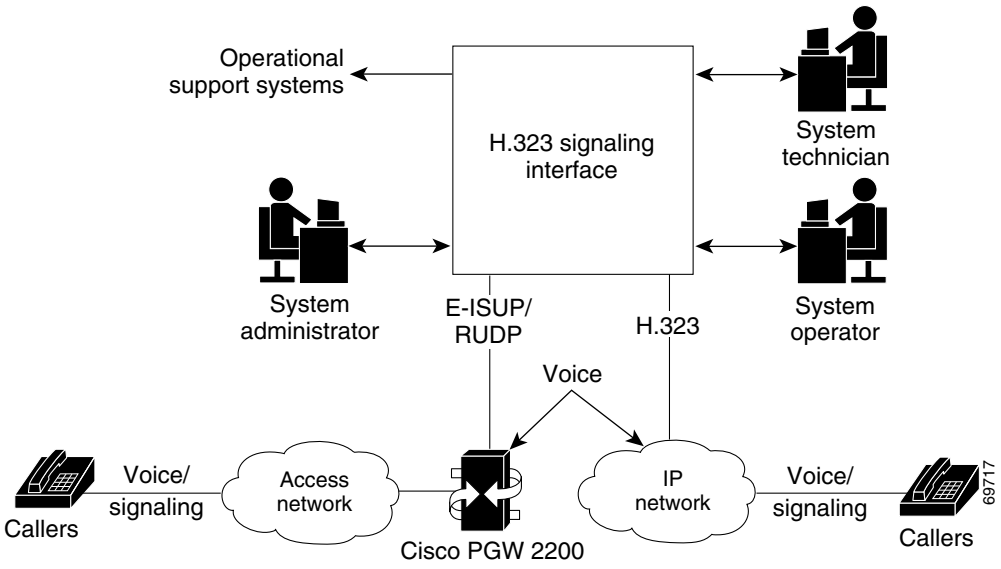
The Cisco HSI adds an H.323 interface to the Cisco Public Switched Telephone Network (PSTN) Gateway (PGW 2200). This interface allows calls to be established between the PSTN and an H.323 network (see Figure 1-1).

The Cisco HSI provides the following services:

- Translation of signaling protocols for establishing, controlling, and releasing calls
- Administration of network parameters and protocol capabilities
- System and call-related statistics
- Fault reporting
- Overload management
- Event logging
- Simple Network Management Protocol (SNMP) interface

The Cisco HSI does not operate in an active/standby configuration and, therefore, does not provide the same level of redundancy as the PGW 2200, which is configured as active/standby. We therefore recommend that you use enough HSI nodes to support the number of simultaneous calls plus one. This ensures (Trunk Group Caveats dependant) that, if one HSI fails, the calls are still adequately supported by the remaining active HSIs.

Figure 1-1 Cisco HSI System Overview



PGW 2200

The PGW 2200 consists of the hardware and software that perform the signaling and call control tasks (such as digit analysis, routing, and circuit selection) and seamlessly switch calls from the PSTN through to the IP network.

IP Network

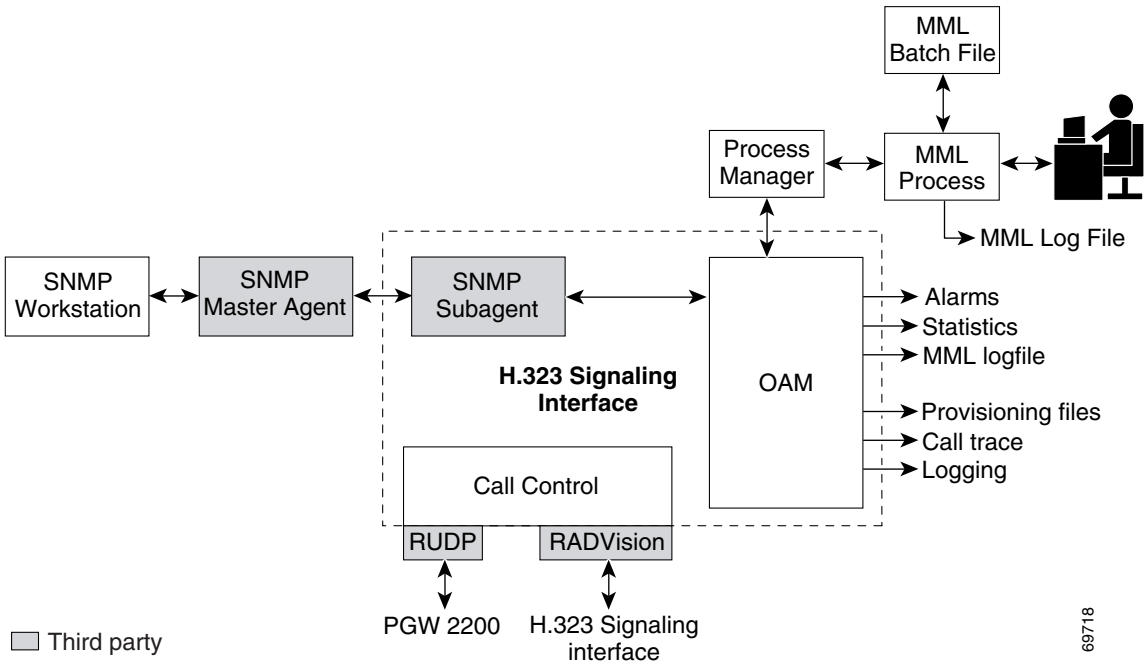
The purpose of the Cisco HSI is to enable the PGW 2200 to interoperate with the H.323 network.

Cisco HSI System Description

The Cisco HSI system has two subsystems (see Figure 1-2):

- Operations, Administration, and Maintenance (OAM) subsystem
- Call control subsystem

Figure 1-2 Cisco HSI Subsystems



OAM Subsystem

The OAM subsystem provides the following services:

- Man-Machine Language (MML) interface that enables you to retrieve operational parameters and modify configuration values through direct input or through batch files
- SNMP interface that allows statistics and alarm retrieval
- Management to provide automatic restart of the Cisco HSI application and control over the running of the process
- Statistics, events, call trace, and alarm output to files
- Alarm events output to the MML interface
- Overload control

Call Control Subsystem

The call control subsystem provides the following services:

- Manages the Reliable User Data Protocol (RUDP) and H.323 stacks
- Implements Enhanced ISDN User Part (E-ISUP)
- Manages H.323 call control
- Performs the conversion of calls between H.323 and E-ISUP call control messages
- Provides call management and overload reduction actions

RUDP

RUDP transports the E-ISUP messages between the PGW 2200 and the Cisco HSI.

RUDP is a Cisco proprietary, connection-oriented, packet-based transport protocol.

RADVision H.323

The Cisco HSI 4.1 release uses the RADVision 4.1 H.323 stack. The HSI uses the H.225 (Q.931 and registration, admission, and status [RAS] protocol) and H.245 protocols to implement the H.323 gateway signaling function.

RADVision H.323 enables the creation of real-time voice H.323 calls over IP networks.

E-ISUP

E-ISUP is a proprietary Cisco protocol based on ISUP. E-ISUP is used for inter-PGW 2200 call control. E-ISUP uses a subset of ISUP messages. The main differences between ISUP and E-ISUP are as follows:

- E-ISUP is for the control of packet voice connection. It does not have circuit management messages such as circuit reset and blocking.
- E-ISUP is transported over RUDP in an IP network.
- E-ISUP enables PGW 2200s to transport Session Description Protocol (SDP) information (such as endpoint IP address and codec specifications) for call endpoints.

The Cisco HSI provides a conversion between the E-ISUP call control protocol originating from the PGW 2200 and the H.323 call control protocol originating from the IP network (see Figure 1-1).

New Features in Cisco HSI Release 4.1

The features introduced in the Cisco HSI 4.1 release are:

- Support of H.323 Version 4
- Invocation of Empty Capabilities Set
- Notify Support
- Disk Mirroring
- Additional security features

Operational Environment

This section provides operational environment requirements for the Cisco HSI.

Hardware Requirements

The hardware requirements for the Cisco HSI are documented in the *Cisco Media Gateway Controller Hardware Installation Guide*. See the section “Cisco MGC Host Platforms” in Chapter 1.

Software Requirements

The software requirements for the Cisco HSI are documented in the *Cisco Media Gateway Controller Software Release 9 Installation and Configuration Guide*.

Security

The application does not directly provide security features. All security must be implemented at the UNIX level.

Cisco HSI Recovery

The Cisco HSI automatically restarts the main application process if that process terminates.

Note If the system is rebooted, the HSI is not started automatically unless the HSI was already activated prior to the reboot.

Cisco HSI System Limitations

The Cisco HSI does not implement security features.

Note You cannot run the Cisco HSI on the same hardware platform with the Cisco PGW.

Installing and Configuring Cisco HSI Software

Introduction

This chapter contains instructions for installing and configuring the Cisco H.323 Signaling Interface (HSI). This chapter contains the following sections:

Hardware and Software Requirements

The hardware requirements for the Cisco HSI are documented in the *Cisco Media Gateway Controller Hardware Installation Guide*. See the section "Cisco MGC Host Platforms" in Chapter 1.

The software requirements for the Cisco HSI are documented in the *Cisco Media Gateway Controller Software Release 9 Installation and Configuration Guide*.

Installing the Operating System

The appropriate operating system must be installed before you install the Cisco HSI. Instructions for installing the operating system on the appropriate platform are in the *Cisco Media Gateway Controller Software Release 9 Installation and Configuration Guide,* located at the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/access/sc/rel9/swinstl/index.htm

After completing the operating system installation, return to this document for Cisco HSI installation procedures.

Installing the Cisco HSI

This section provides step-by-step instructions for installing the Cisco HSI.

Before You Start

Complete the preinstallation tasks listed in Table 2-1 before installing the Cisco HSI. Use the checklist to ensure that each task is completed. Detailed instructions for completing some tasks follow the checklist.

Table 2-1 Preinstallation Tasks Checklist

Check	Preinstallation Task
	Ensure that the required operating system is installed on the appropriate hardware platform.
	Configure group and user names, as described in the "Configuring Groups and Users" section on page 2-2.
	Gather the information listed in Table 2-2 and note it in the table for reference during the installation.
	Have your company internal support information and Cisco support contact information readily available so you can get help with the installation if needed. If you have questions or need assistance, see the "Obtaining Technical Assistance" section on page 18.

Configuring Groups and Users

You must configure groups and users for the Cisco HSI on each host server. A user must be a member of the "mgcgrp" group to use certain Cisco HSI functions, such as Man-Machine Language (MML).

To configure groups and users, complete the following steps:

Step 1 Log in as root.

Step 2 At the # prompt, enter the following commands:

 # mkdir -p /export/home/users/mgcusr

 # mkdir /export/BUILDS

 # mkdir /export/PATCHES

 # cd /export/home/users

 # groupadd -g 20000 mgcgrp

 # useradd -u 20001 -g 20000 -d /export/home/users/mgcusr -s /bin/csh mgcusr

 # chown mgcusr:mgcgrp mgcusr

 # passwd mgcusr <*type password twice*>

(Enter and confirm password)

Step 3 Log out, then log in as **user mgcusr,** using the password you applied in Step 2.

Step 4 Verify that you are in directory /export/home/users/mgcusr by entering the following command:

 # pwd

Step 5 Enter the following command:

 # vi .cshrc

Step 6 Enter the vi insert mode by entering the following command:

i (enter insert mode)

Step 7 Enter the following text on the first line:

source /opt/GoldWing/currentPM/local/setup.gw.csh

Step 8 Save the file and quit vi by entering the following commands:

[**Esc**] (exit insert mode)

:wq (write file and quit)

Step 9 Enter the following command:

 # chmod 777 .cshrc

Cisco HSI Installation Information

Gather the information listed in Table 2-2 before you begin the Cisco HSI installation. Use the Notes column in this table to record the information. Several steps in the installation procedure require you to provide this information. Refer to this table as you proceed through the Cisco HSI installation steps.

Table 2-2 Cisco HSI Installation Information

Required Information	Notes
Base directory path	**Note** We strongly recommend that you accept the default base directory path.
Cisco HSI user name	Default: mgcusr
Cisco HSI group name	Default: mgcgrp
Gatekeeper IP address	
Gatekeeper port	Default: 1719
Gateway prefix	
Terminal alias	
Gatekeeper ID	**Note** This ID must match the entry configured in the gatekeeper.
E-ISUP host port	**Note** Typically 8003, but this entry must match the peer port setting of the IPLNK object in the PGW 2200 configuration.
VSC1[1] name (either the DNS[2] host name, if DNS is configured, or the IP address of the Cisco PGW 2200[3])	
VSC1 port	**Note** Typically 8003, but this entry must match the peer port setting of the IPLNK object in the PGW 2200 configuration.
Installation node ID	

Table 2-2 Cisco HSI Installation Information (continued)

Required Information	Notes
Hardware platform	
Installation location	

1. VSC = virtual switch controller
2. DNS = domain name system
3. PGW = PSTN Gateway

The Cisco HSI application is distributed as a tar file (with filename GoldWing-*xxxx*.tar in which *xxxx* is the version ID, for example, GoldWing-4.1.tar) or as a CD-ROM.

The default installation directory is /opt/GoldWing. We recommend that you install the software at the default location. More than one version of the software can exist within subdirectories, for example /opt/GoldWing/4.1

Links point to the currently active version of the Cisco HSI application, as follows:

- currentPM points to the current version to use for all software except the call processing application.
- currentGW points to the version that may not be the latest version of the call processing application. (GWmain)

Table 2-3 shows the subdirectories of the /opt/GoldWing/currentPM directory.

Table 2-3 CurrentPM Subdirectories

Subdirectory	Contents
./bin	All compiled executables.
./local	All scripts.
./etc	Base configuration files.
./lib	Shared libraries required by executables.
./toolkit	Toolkit files.
./var	Volatile directory that contains file locks and so on.
./var/log	Default log directory.
./var/prov	Provision system writes provisioning config files here.
./var/trace	Trace logs are written here.

Exported provisioning files are stored in /opt/GoldWing/export.

Installing Cisco HSI

This section provides step-by-step instructions for installing a single Cisco HSI for use with a simplex PGW 2200 configuration (a configuration with one Cisco PGW 2200 host). To install a dual Cisco HSI for use with a redundant PGW 2200 configuration (a configuration with two Cisco PGW 2200 hosts), complete the steps in this section and then proceed to the "Installing Multiple Cisco HSIs in a Redundant PGW 2200 Configuration" section on page 2-10.

Note In the following installation procedure, the package name is OTTgw000 and the version of the software is 4.1; the /export/BUILDS directory is used to install the system software.

To install the Cisco HSI, complete the following steps:

Step 1 Verify that the operating system is installed. See the "Installing the Operating System" section on page 2-1 for more information.

Step 2 Login as root.

Step 3 Issue the command: **cd /export**

Step 4 The initial step for downloading the HSI software depends upon the media from which you obtain the software:

- If you download the software from a server, it will be in a tar file. Issue the following command:

  # tar xvf GoldWing-4.1.tar

  This command displays the following text:

```
x ./4.1/APPLICATIONS, 0 bytes, 0 tape blocks
x ./4.1/APPLICATIONS/OTTgw000.pkg, 38954496 bytes, 76083 tape blocks
x ./4.1/install.sh, 5223 bytes, 11 tape blocks
x ./4.1/uninstall.sh, 3053 bytes, 6 tape blocks
```

  Note The byte and block counts for your installation may be different from those provided in the preceding example.

- If you download the software from a CD-ROM, insert the Cisco HSI 4.1 CD-ROM into the drive and issue the following commands:

  # mkdir BUILDS/4.1
  # cp -r /cdrom/hsi_4.1* /* /export/BUILDS/4.1

Step 5 At the # prompt, enter the following commands:

 # cd /export/BUILDS/4.1
 # ./install.sh

The following text displays:

```
Processing package instance <OTTgw000> from </export/BUILDS/4.1/APPLICATIONS/OTTgw000.pkg>
GoldWing H323 Adjunct Processor V0.1.6
(sparc) 4.1
Copyright (c) 2001 Cisco Systems, Ltd.
All Rights Reserved
This product is protected by copyright and distributed under
licenses restricting copying, distribution and decompilation.
Enter GoldWing base directory path (default /opt/GoldWing) [?,q]
```

Step 6 Press **Enter** to select the default HSI base directory path.

Caution We strongly recommend that you select the default base directory path. Operational issues might arise if other directories are used.

The following text displays:

```
Enter base directory path (default /opt/GoldWing/4.1) [?,q]
```

Step 7 Press **Enter** to select the default base directory path. The following text displays:

```
Enter GoldWing user name
```

Step 8 Type the Cisco HSI user name **mgcusr** and press **Enter** (the default user name is cisco). The following text displays:

```
Enter GoldWing group name
```

Step 9 Type the Cisco HSI group name **mgcgrp** and press **Enter** (the default user group name is sysadmin). The following text displays:

```
Enter GateKeeper IP Address
```

Step 10 Type the gatekeeper IP address (see Table 2-2) and press **Enter**. The following text displays:

```
Enter GateKeeper Port
```

Step 11 Type the gatekeeper port (see Table 2-2) and press **Enter** (the default port is 1719). The following text displays:

```
Enter GateWay Prefix
```

Step 12 Type the gateway prefix (see Table 2-2) and press **Enter**.

Note The gateway prefix is the prefix that, when dialed from the H.323 network, causes the Cisco HSI to route the call over E-ISUP to the PGW 2200.

The following text displays:

```
Enter Terminal Alias
```

Step 13 Type the terminal alias (see Table 2-2) and press **Enter**. The following text displays:

```
Enter GateKeeper Id
```

Step 14 Type the gatekeeper ID (see Table 2-2) and press **Enter**.

Note The gatekeeper ID must match the entry configured in the gatekeeper.

The following text displays:

```
Enter E-ISUP Host Port
```

Step 15 Type the E-ISUP host port (see Table 2-2) and press **Enter**.

Note The E-ISUP host port is typically 8003, but it must match the peer port setting of the IPLNK object in the PGW 2200 configuration.

The following text displays:

```
Enter VSC1 Name
```

Step 16 Type the VSC1 name and press **Enter**.

Note The VSCI name is either the DNS host name (if DNS is configured) or the IP address of the PGW 2200.

The following text displays:

```
Enter VSC1 Port
```

Step 17 Type the VSC1 port number (see Table 2-2) and press **Enter**.

Note The VSCI port is typically 8003, but it must match the port setting of the IPLNK object in the PGW 2200 configuration.

The following text displays:

```
Enter Installation NodeId
```

Step 18 Type the installation node ID (see Table 2-2) and press **Enter**.

Note The installation node ID is a text field typically used by network designers for identification purposes. Entering a value in this field does not affect functionality.

The following text displays:

```
Enter Hardware Platform
```

Step 19 Type the hardware platform name (see Table 2-2) and press **Enter** (typically, accept the default platform name). The following text displays:

```
Enter Installation Location
```

Step 20 Type the installation location (see Table 2-2) and press **Enter**.

Note The installation location field is a text field typically used by network designers for identification purposes. Entering a value in this field does not affect functionality.

The following is an example of the screen that displays:

```
## Executing checkinstall script.
Modified Environment is:
-------------------------
BASEDIR=/opt/GoldWing/4.1
GWHOME=/opt/GoldWing
GWUSR=mgcusr
GWGRP=mgcgrp
GWCONF_IP="10.70.54.53"
GWCONF_PORT="1719"
GWCONF_PREFIX="0208"
GWCONF_ALIAS="cisco@OuterLondonDomain.com"
GWCONF_GKID="OuterLondon"
GWCONF_HOST_PORT=8003
GWCONF_VSC1_NAME=goliath
GWCONF_VSC1_PORT=8003
GWCONF_NODEID="H323-GW1"
GWCONF_HARDWARE="Sun Netra T1"
GWCONF_LOCATION="H323 - GW1"
-------------------------
The selected base directory </opt/GoldWing/4.1> must exist before installation is
attempted.
Do you want this directory created now [y,n,?,q]
```

Step 21 Type **y** to create the version directory. The following text displays:

```
Using </opt/GoldWing/4.1> as the package base directory.
## Processing package information.
## Processing system information.
## Verifying disk space requirements.
## Checking for conflicts with packages already installed.
## Checking for setuid/setgid programs.
This package contains scripts which will be executed with super-user
permission during the process of installing this package.
Do you want to continue with the installation of <OTTgw000> [y,n,?]
```

Step 22 Review the output before you continue the installation. Type **y** to continue. The files are installed. The following text displays:

```
Installing GoldWing H323 Adjunct Processor V0.1.6 as <OTTgw000>
    ## Installing part 1 of 1.
    /etc/init.d/CiscoGW
    /opt/GoldWing/4.1/bin/GWmain
    /opt/GoldWing/4.1/bin/PMmain
    /opt/GoldWing/4.1/bin/mml
    /opt/GoldWing/4.1/bin/msg.conf
    /opt/GoldWing/4.1/bin/parse
    /opt/GoldWing/4.1/etc/GWmain.base.conf
    /opt/GoldWing/4.1/etc/GWmain.default.conf
    /opt/GoldWing/4.1/etc/GWmain.static.conf
    /opt/GoldWing/4.1/etc/H323SkeletonFileSimple.dat
    /opt/GoldWing/4.1/etc/parse.exclude.list
    /opt/GoldWing/4.1/etc/parse.list
    /opt/GoldWing/4.1/lib/libgwMib_shlib.so
    /opt/GoldWing/4.1/var/prov/active_config <symbolic link>
    [ verifying class <none> ]
```

```
[ verifying class <script> ]
## Executing postinstall script.
Installed package instance is: OTTgw000
Installation of <OTTgw000> was successful.
Installed package instance environment variables are:
----------------------------------------------------
PKGINST=OTTgw000
VERSION=4.1
BASEDIR=/opt/GoldWing/4.1
GWHOME=/opt/GoldWing
MGCUSR=mgcusr
MGCGRP=mgcgrp
----------------------------------------------------
Setting link /opt/GoldWing/currentPM.
Setting link /opt/GoldWing/currentGW.
```

Installation of the Cisco HSI is now complete. The directory /opt/GoldWing now displays as follows:

```
drwxr-xr-x   7 cisco    sysadmin     512 Jan  9 18:31 4.1
lrwxrwxrwx   1 cisco    sysadmin      19 Jan  9 18:31 currentGW -> /opt/GoldWing/4.1
lrwxrwxrwx   1 cisco    sysadmin      19 Jan  9 18:31 currentPM -> /opt/GoldWing/4.1
-rwxrwxr-x   1 root     other       3053 Jan  9 18:31 uninstall.sh
```

Note The links currentPM and currentGW point to the currently active version of the Cisco HSI. The uninstall script has been copied here for convenience, but it can be run only by root user.

To check the Cisco HSI installation, enter **pkgchk OTTgw000**.

Note The **pkgchk** command reports File size / Checksum information. This information may suggest errors because the post-installation scripts modify some of the files with user configuration information for which the user was prompted during the installation procedure. These messages are expected and do not indicate a problem with the installation.

Note The package name is OTTgw000. If more than one instance of the package is installed, the package name has a suffix (for example, OTTgw000.2, OTTgw000.3, and so on).

Outside of the /opt/GoldWing directory, the start/stop script CiscoGW is copied to the /etc/init.d directory.

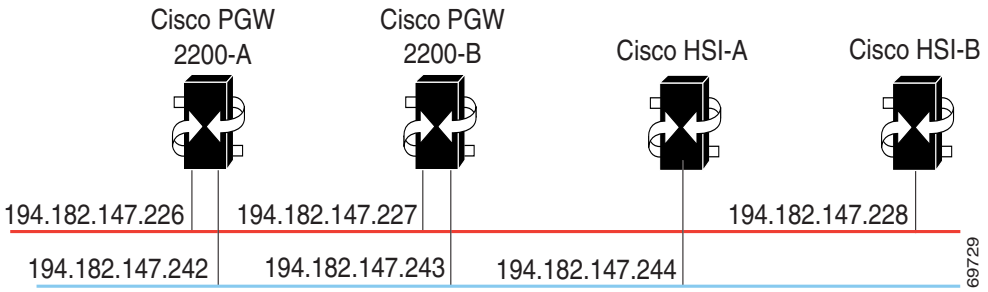
When the installation is complete, a file named PKINST is written to the base directory on the installed software.

Caution Do not modify the PKINST file. It contains information derived from the installation, and the uninstall script uses the PKINST file in the version directory to determine which package name to remove if more than one instance of the package is installed.

Installing Multiple Cisco HSIs in a Redundant PGW 2200 Configuration

This section describes how to install and configure two Cisco HSI for use with a redundant Cisco PGW 2200 configuration (See Figure 2-1).

Figure 2-1 Dual Cisco HSI with a Redundant PGW 2200 Configuration



Caution To ensure the successful installation of two Cisco HSIs, after Step 22 of the "Installing Cisco HSI" section on page 2-5, provision the software for the active host first before proceeding to Step 1 below. See "Configuring the Cisco HSI" section on page 2-12 for configuration information.

Only one active provisioning session is permitted, and provisioning is permitted only on the active Cisco HSI.

Exit the provisioning session on the active host and continue to Step 1 below. If software is not provisioned after it is installed on the active host, the stand-by host is not synchronized with the active host. As a result, a forced switchover might fail.

To install two Cisco HSIs for a redundant PGW 2200 configuration (a configuration with two Cisco PGW 2200 hosts), complete the following steps:

Step 1 Continuing from Step 22 of the "Installing Cisco HSI" section on page 2-5, exit server 1.

Step 2 Log in to server 2 as root and go to the # prompt.

Step 3 Insert the Cisco HSI CD-ROM in the CD-ROM drive.

Step 4 Follow the installation instructions found in Step 3 through Step 22 of the "Installing Cisco HSI" section on page 2-5.

Installation of the dual Cisco HSI for a redundant PGW 2200 configuration is now complete. (See the example configuration script in the following section.)

Dual HSI Example Configuration Script

The following example script configures the network topology depicted in Figure 2-1.

Example

```
HSI-A (Blue network)
prov-add:name="SYS_CONFIG_STATIC",HOST_PORT_NUMBER1="9001"
prov-add:name="SYS_CONFIG_STATIC",HOST_PORT_NUMBER2="0"
prov-add:name="SYS_CONFIG_STATIC",VSCA_IPADDR1="194.182.147.242"
prov-add:name="SYS_CONFIG_STATIC",VSCA_IPADDR2="194.182.147.242"
prov-add:name="SYS_CONFIG_STATIC",VSCA_PORT_NUMBER1="8003"
prov-add:name="SYS_CONFIG_STATIC",VSCA_PORT_NUMBER2="8003"
prov-add:name="SYS_CONFIG_STATIC",VSCB_IPADDR1="194.182.147.243"
prov-add:name="SYS_CONFIG_STATIC",VSCB_IPADDR2="194.182.147.243"
prov-add:name="SYS_CONFIG_STATIC",VSCB_PORT_NUMBER1="8003"
prov-add:name="SYS_CONFIG_STATIC",VSCB_PORT_NUMBER2="8003"

HSI-B (Red network)
prov-add:name="SYS_CONFIG_STATIC",HOST_PORT_NUMBER1="9002"
prov-add:name="SYS_CONFIG_STATIC",HOST_PORT_NUMBER2="0"
prov-add:name="SYS_CONFIG_STATIC",VSCA_IPADDR1="194.182.147.226"
prov-add:name="SYS_CONFIG_STATIC",VSCA_IPADDR2="194.182.147.226"
prov-add:name="SYS_CONFIG_STATIC",VSCA_PORT_NUMBER1="8004"
prov-add:name="SYS_CONFIG_STATIC",VSCA_PORT_NUMBER2="8004"
prov-add:name="SYS_CONFIG_STATIC",VSCB_IPADDR1="194.182.147.227"
prov-add:name="SYS_CONFIG_STATIC",VSCB_IPADDR2="194.182.147.227"
prov-add:name="SYS_CONFIG_STATIC",VSCB_PORT_NUMBER1="8004"
prov-add:name="SYS_CONFIG_STATIC",VSCB_PORT_NUMBER2="8004"
```

Starting the Cisco HSI

To start the Cisco HSI, execute the start script as the root user and enter the following command:

 # /etc/init.d/CiscoGW start

Note The application runs as root user because this is a requirement of a Simple Network Management Protocol (SNMP) subagent application. If you do not run this script as the root user, the SNMP subagent fails to connect to the master agent.

Stopping the Cisco HSI

To stop the Cisco HSI, log in as root user and enter the following command:

 # /etc/init.d/CiscoGW stop

Note The command to stop the Cisco HSI returns information such as the following:

```
/etc/init.d/CiscoGW stop
Signalling PMmain to shut down
Signalling GWmain[739] to shut down
Process 'PMmain' not found
Process 'GWmain' not found
...shutdown complete
```

The "not found" messages do not indicate a problem. The shutdown script first attempts to shutdown the HSI processes gracefully (kill -39). The script then checks to determine whether the processes still exist. If HSI processes remain active, the script uses the kill -9 command. The "shutdown complete" announcement indicates that the shutdown script succeeded in stopping the HSI.

Configuring the Cisco HSI

To configure the Cisco HSI, you must first access the user interface. Use the **mml** command (see Appendix A, "MML User Interface and Command Reference" for more information). If the setup.gw file has been sourced, it is in the user path. Use the provisioning commands to configure the Cisco HSI as required (see Chapter 3, "Provisioning the Cisco HSI" and Appendix A, "MML User Interface and Command Reference" for more information).

Upgrading the Cisco HSI

Before removing an old version of the Cisco HSI, install the new version of the software. You can export a provisioning session to a flat file in a format that can be used as input to another provisioning session (see prov-exp in Appendix A, "MML User Interface and Command Reference," for more information).

Note To upgrade to Cisco HSI 4.1, if you have not partitioned disks according to the information provided in the partitioning tables presented in the *Cisco Media Gateway Controller Software Release 9 Installation and Configuration Guide*, you must repartition the disks and reinstall the operating system.

When you upgrade the Cisco HSI version, the following conditions apply:

- The Cisco HSI must first be stopped before installation is allowed to proceed.
- The installation of the new software does not overwrite the existing installed version.
- The installation of a new version results in a new version directory being created in the /opt/GoldWing parent directory. The links currentPM and currentGW are updated to point to this new version.

- The system should be restarted to enable the re-initialization of the SNMP processes. The craftsperson needs to change the user to root and issue the following UNIX command:

```
shutdown -i 6 -g 0 -y
```

Note To revert to a previous version of the software, manually modify the currentPM and currentGW links in the/opt/GoldWing parent directory to point to the previous version.

The uninstall.sh script uses the PKINST file in the version directory to determine which package name to remove.

Caution When upgrading the Cisco HSI, you must install the security package CSC0h013 before you remove (uninstall) the preceding version of the HSI software. If you do not, the CSC0h013 security package will not operate. This step is not required if you are performing a fresh installation of the Cisco HSI.

Removing the Cisco HSI

To remove the Cisco HSI, complete the following steps:

Step 1 Log in as root.

Step 2 Enter the following command to stop the Cisco HSI:

 # /etc/init.d/CiscoGW stop

Step 3 Enter the following commands:

 # cd /opt/GoldWing

 # ls -l

The following is an example of the screen that displays:

```
drwxr-xr-x   7 cisco    sysadmin     512 Jan  9 18:31 4.1
lrwxrwxrwx   1 cisco    sysadmin      19 Jan  9 18:31 currentGW -> /opt/GoldWing/4.1
lrwxrwxrwx   1 cisco    sysadmin      19 Jan  9 18:31 currentPM -> /opt/GoldWing/4.1
-rwxrwxr-x   1 root     other       3053 Jan  9 18:31 uninstall.sh
```

Step 4 Enter the **uninstall** command and specify the version of the software that you want to uninstall, for example:

 # ./uninstall.sh 4.1

The following text displays:

```
Warning: This script will remove the package OTTgw000
Do you wish to proceed? [n] [y,n,?,q]
```

Step 5 Type **y** and press **Enter.** The following text displays:

```
Deleting generated files in /opt/GoldWing/4.1
The following package is currently installed:
OTTgw000        GoldWing H323 Adjunct Processor V0.1.6
(sparc) 4.1
Do you want to remove this package?
```

Step 6 Type **y** and press **Enter**. The following text displays:

```
## Removing installed package instance <OTTgw000>
This package contains scripts which will be executed with super-user
permission during the process of removing this package.
Do you want to continue with the removal of this package [y,n,?,q]
```

Step 7 Type **y** and press **Enter**. The following text displays:

```
## Verifying package dependencies.
## Processing package information.
## Executing preremove script.
## Removing pathnames in class <script>
/opt/GoldWing/4.1/local/setup.gw
/opt/GoldWing/4.1/local/pmStart.sh
/opt/GoldWing/4.1/local/gwhalt
/opt/GoldWing/4.1/local/CiscoGW
## Removing pathnames in class <none>
/opt/GoldWing/4.1/local
/opt/GoldWing/4.1/lib/libgwMib_shlib.so
/opt/GoldWing/4.1/lib
/opt/GoldWing/4.1/etc/parse.list
/opt/GoldWing/4.1/etc/parse.exclude.list
/opt/GoldWing/4.1/etc/H323SkeletonFileSimple.dat
/opt/GoldWing/4.1/etc/GWmain.static.conf
/opt/GoldWing/4.1/etc/GWmain.request.conf
/opt/GoldWing/4.1/etc/GWmain.default.conf
/opt/GoldWing/4.1/etc/GWmain.conf
/opt/GoldWing/4.1/etc/GWmain.base.conf
/opt/GoldWing/4.1/etc
/opt/GoldWing/4.1/bin/parse
/opt/GoldWing/4.1/bin/msg.conf
/opt/GoldWing/4.1/bin/mml
/opt/GoldWing/4.1/bin/PMmain
/opt/GoldWing/4.1/bin/GWmain
/opt/GoldWing/4.1/bin
/opt/GoldWing/4.1/PKGINST
/etc/init.d/CiscoGW
/etc/init.d <shared pathname not removed>
/etc <shared pathname not removed>
## Executing postremove script.
## Updating system information.

Removal of <OTTgw000> was successful.
```

Provisioning the Cisco HSI

Introduction

This chapter describes the data that must be provisioned for the Cisco H.323 Signaling Interface (HSI). The data is divided into two areas: system configuration and H.323 stack data. This chapter contains the following sections:

Cisco HSI Configuration

All configuration data is contained within configuration files. Cisco HSI starts with an initial configuration file in $GWHOME/currentGW/etc/GWmain.conf. This file is created during installation of the software.

The configuration data within the file is defined as dynamic, static, or constant:

- Dynamic data can be modified by a provisioning session (see Appendix A, "MML User Interface and Command Reference"). It can be activated on the currently running Cisco HSI.
- Static data can be modified by a provisioning session but cannot be activated on a running Cisco HSI. Changes to dynamic and static data can be written to a separate provisioning file (in $GWHOME/currentGW/var/prov/configname/session.dat) that can be used during subsequent restarts of the Cisco HSI.
- Constant configuration data is contained within the configuration file and cannot be modified by provisioning sessions. Constant configuration data can be modified only by system technicians or administrators who use UNIX editing tools. This data is replicated from the initial configuration file into the provisioning files, and is included in subsequent provisioning sessions.

  Examples of the use of constant data are given in Appendixes D, E, F, and G. These appendixes determine the mapping of cause values for incoming and outgoing H.323 and Enhanced ISDN User Part (E-ISUP) messages. System technicians can modify these values in the initial configuration file to explicitly choose the mappings for their system.

When a provisioning session creates a new configuration file, it also verifies that provisioned data is within allowable ranges and indicates this in the start of the file. It checksums the configuration file and writes the checksum as $GWHOME/currentGW/var/prov/configname/checksum.dat. When the Cisco

HSI starts up, it attempts to read the active configuration, checks that the configuration has been verified, and ensures that the checksum matches. If the active configuration is not verified or if the checksum is faulty, the configuration reverts to using the $GWHOME/currentGW/etc/GWmain.conf file.

All configuration data that can be set in the system is defined in the Skeleton Configuration file (see Appendix B, "Skeleton Configuration File"). The Skeleton Configuration file defines the data names and types (strings or numbers), and defines whether the data is dynamic, static, or constant.

MML Configuration Commands

There are three types of MML configuration command:

- Configuration session commands that work with entire provisioning data files (see Table 3-1)
- Configuration component or parameter commands that perform actions on components or parameters affecting a specific data file (see Table 3-2)
- Configuration export commands

For more information about MML configuration commands, see Appendix A, "MML User Interface and Command Reference."

Note Parameter names used in MML commands are not case sensitive.

Table 3-1 Configuration Session Commands

Command	Description
prov-sta	Starts a provisioning session to create a new configuration or modify an existing configuration
prov-cpy	Activates the configuration settings in the current provisioning session
prov-stp	Terminates the provisioning session and saves the configuration

Table 3-2 Configuration Component or Parameter Commands

Command	Description
prov-add	Adds a component to the Cisco HSI
prov-dlt	Deletes a provisioned component
prov-ed	Modifies a provisioned component
prov-rtrv	Retrieves information about an existing provisioning session

The configuration export command is **prov-exp**, which exports the currently provisioned configuration of the Cisco HSI to a file.

Introduction to MML Command Operation for HSI

After the HSI software is installed, you can configure additional items. The following MML command examples show how to enable DTMF capability on the HSI. (For a description of the sys_config_static entry and the dtmf parameters, please see the section (System Configuration Data).

Initiating an MML Session to Enable DTMF on the HSI

The following MML command example shows how to start an MML session and enable DTMF support of the HSI:

Step 1 As root user, issue the following command:

```
/etc/init.d/CiscoGW start
```

Step 2 As mgcusr, begin an MML session by issuing the following command:

mml

Step 3 To enable DTMF support on the HSI, issue the following set of commands:

```
prov-sta:srcver=active, dstver=myconf
```

Note The preceding command creates a new configuration, based on the current configuration, called myconf.

```
prov-add:name=sys_config_static, dtmfsupportedtype=dtmf
prov-add:name=sys_config_static, dtmfsupporteddirection=both
prov-cpy
restart-softw
```

Note Certain configuration changes do not take effect until the HSI is restarted. After the **restart-softw** command is issued, the HSI restarts in approximately 20 seconds.

Caution Use MML commands to perform all HSI configuration. Never manually edit system configuration files because they do not undergo the same parse checks as MML commands. In addition, the HSI uses a machine-generated checksum to verify the system files. If you modify the system configuration files manually, the HSI cannot use them and reverts to the base configuration.

Verifying the Configuration

The following MML command examples show how to verify that configuration changes have been correctly processed:

Step 1 To retrieve information about the current provisioning session, issue the following command:

```
prov-rtrv:list
```

Note The HSI prints an asterisk next to the currently active configuration.

Step 2 To display the entire configuration, issue the following command:

```
rtrv-config
```

To display a subset of the configuration, one can issue a command such as the following:

```
rtrv-config:sys_config_static
```

Step 3 To exit the MML command interpreter, issue the following command:

```
quit
```

Reverting to the Base Configuration

The following MML command examples show how to revert to the base HSI configuration:

Step 1 To begin an MML session, issue the following command:

```
mml
```

Step 2 To revert to the base HSI configuration, issue the following command:

```
restart-softw:init
```

Note The **restart-softw:init** command is derived from the initial installation script. (See Step 6 in the "Installing Cisco HSI" section on page 2-5.) To return to the configuration "myconf," one would issue the command **restart-softw:myconf**.

System Configuration Data

System configuration data can be static or dynamic. Static data can be activated only at startup. Dynamic data can be activated during system run time.

Static System Data

To modify the static system data parameters in Table 3-3, use the **sys_config_static** MML name variable for the **prov-add**, **prov-dlt**, and **prov-ed** commands. Stop and restart the application for the changes to take effect.

In the following example, the **prov-add** command adds the static system data parameter VSCA_PORT_NUMBER1 to a static configuration file. The **prov-ed** command modifies the value of the VSCA_PORT_NUMBER1 parameter. The **prov-dlt** command deletes the VSCA_PORT_NUMBER1 parameter from the static configuration file.

Example

```
prov-add:name=sys_config_static,vsca_port_number1=8003
prov-ed:name=sys_config_static,vsca_port_number1=8002
prov-dlt:name=sys_config_static,vsca_port_number1
```

The parameters in Table 3-3 are written to a static configuration file or to a section within a file.

Table 3-3 Static System Data Parameters

Parameter	Type	Description
HOST_PORT_NUMBER1	[0-65535]	The first port number to be used by the Cisco HSI. The default value is 0. **Note** This value must match the peer port setting on the PGW[1] 2200 E-ISUP IPLNK object.
HOST_PORT_NUMBER2	[0-65535]	The second port number to be used by the Cisco HSI. The default value is 0. **Note** This value should always be set to 0.
VSCA_IPADDR1	STRING	The primary IP address of the primary PGW 2200.
VSCA_IPADDR2	STRING	The secondary IP address of the primary PGW 2200. **Note** This value must match that of VSCA_IPADDR1.
VSCB_IPADDR1	STRING	The primary IP address of the secondary PGW 2200. **Note** This parameter is not used in a standalone PGW configuration.
VSCB_IPADDR2	STRING	The secondary IP address of the secondary PGW 2200. **Note** The value of this parameter must match that of VSCB_IPADDR1. This parameter is not used in a standalone PGW configuration.
VSCA_PORT_NUMBER1	[0-65535]	The first port number of the primary PGW 2200.
VSCA_PORT_NUMBER2	[0-65535]	The second port number of the primary PGW 2200. **Note** This value must match that of VSCA_PORT_NUMBER1.
VSCB_PORT_NUMBER1	[0-65535]	The first port number of the secondary PGW 2200. **Note** This parameter is not used in a standalone PGW configuration.
VSCB_PORT_NUMBER2	[0-65535]	The second port number of the secondary PGW 2200. **Note** The value of this parameter must match that of VSCA_PORT_NUMBER2. This parameter is not used in a standalone PGW configuration.
ClipClirSupported	STRING	CLI Presentation or restriction is enabled if this parameter is present and set to anything other than “”. For example, to enable CLIP/CLIR support, set this parameter explicitly to “Enabled.”
RaiSupported	STRING	RAI support is enabled if this parameter is present and set to anything other than “”. For example, to enable RAI support, set this parameter to “Enabled.”
DtmfSupportedDirection	STRING	This is set to “both”, “tx,” or “rx”. If this parameter is not present or is set to any value other than “both,” “tx,” or “rx,” the DTMF Relay feature is disabled.
DtmfSupportedType	STRING	This is set to “dtmf” or “basicString.” If this parameter is not present or set to any other value, the DTMF Relay feature is disabled.
H225PavoSupported	STRING	Pavo support is enabled if this parameter is present and set to anything other than “”. For example, set it to “Enabled.”
PavoRedirScreeningInd	[0-3]	The value of the Pavo redirecting number screening indicator. (If this parameter is not provisioned, the default is Q.931 zero—user provided, not screened.)
PavoRedirReason	[0-15]	The value of the Pavo redirecting number reason field. This parameter has no default. If unprovisioned, the redirecting number parameter will not contain the Reason for Redirection field (octet 3b).

Table 3-3 Static System Data Parameters (continued)

Parameter	Type	Description
PavoRedirPresInd	[0-3]	The value of the Pavo redirecting number presentation indicator. (If this parameter is not provisioned, the default is Q.931 zero—no indication.)
CliInDisplaySupported	STRING	If this parameter is present and set to anything other than “”, the Calling Number is also sent in the DISPLAY IE. The NetMeeting endpoint retrieves the calling party number from the DISPLAY IE in the H.225 setup message. To enable this parameter, set it to “Enabled.”
T38MaxVal	STRING	The T38MaxVal parameter has the following optional attributes that can be assigned values in a specific range. **Note** Values for the following attributes must be expressed in hexadecimal format. • MaxBit—[0x0—0xFFFFFFFF]. Specifies the maximum bit rate in units of 100 bits per second at which a transmitter can transmit or a receiver can receive T.38 FAX data. The default value is 0x90. • FxMaxBuf—[0x0—0xFFFFFFFF]. Specifies the maximum buffer size for the "t38FaxMaxBuffer" parameter for the T.38 over UDP option. The default value is 0xc8. • FxMaxData—[0x0—0xFFFFFFFF]. Specifies the maximum datagram size for the "t38FaxMaxDatagram" parameter for the T.38 over UDP option. The default value is 0x48.
T38Options	STRING	This T.38 Fax parameter is assigned one of the following optional values: • FxFillBit—[0 or 1] The default value is 0. • FxTransMMR—[0 or 1] The default value is 0. • FxRateTransJBIG—[0 or 1] The default value is 0. • FXRate—[Local or Trans] The default value is Trans. • FxUdpEC—[Red or FEC] The default value is Red.
AsymmetricHandlingSupported	STRING	Asymmetric Codec Treatment support is enabled if this parameter is present and set to anything other than “”. To enable Asymmetric Codec Treatment, set this parameter to “Enabled.”
UseConfID	STRING	Use this parameter to specify the precedence of extracting the Global Call ID from the Conference ID or the GUID in the H.225 Setup message. The provisioning of this property to a value other than “” gives precedence to the Conference ID. For example, set it to “Enabled.” To set the precedence to the GUID field, the crafts person can either delete the property from the config or set it to “”.

Table 3-3 Static System Data Parameters (continued)

Parameter	Type	Description
DualCLISupported	STRING	To enable Dual CLI support (see H.246 Annex C), set this parameter to anything other than “”. For example, to explicitly enable Dual CLI support, set this parameter to “Enabled.”
InjectPi8	STRING	If this parameter is set to a text value (for example, “enabled” or “true”), the HSI inserts a progress indicator value of 8 into the H.225 alerting message, which allows creation of a backward speech path. To disable this feature, you can delete the parameter using the command **prov-dlt** or issue the **prov-ed** command and set the value to ““. **Note** Setting the InjectPi8 parameter is required if the PSTN network does not notify the HSI that inband information is available. For instance, when no Optional Backward Call Indicator is present, a backward speech path will not be available.

1. PGW = Public Switched Telephone Network (PSTN) Gateway

Changing Static System Data

To change static system data, you must first determine if it is acceptable to stop currently active calls in 20 seconds. If it is acceptable to stop active calls in 20 seconds, change static system data using the following procedure:

Step 1 Modify the static parameters you want to change.

Step 2 Activate the changed static parameters by issuing the **prov-cpy** command.

Step 3 Issue the command **restart-softw::confirm**.

This command stops the HSI application in 20 seconds and then restarts it. The restarted HSI application reads the changed static system data parameters.

Step 4 To ensure that traffic processing has resumed, issue the command **rtrv-ne-health**.

If you wish to change static system data but it is not acceptable to stop active calls in 20 seconds, use the following procedure:

Step 1 Modify the static parameters you want to change.

Step 2 Activate the changed static parameters by issuing the **prov-cpy** command.

Step 3 Stop call processing by issuing the **stp-callproc** command, specifying the timeout period you require.

Step 4 When the timeout period expires, ensure that all traffic ceased by issuing the command **rtrv-ne-health**.

Step 5 Restart the HSI software by issuing the command **restart-softw**.

This command stops the HSI application and then restarts it. The restarted HSI application reads the changed static system data parameters.

Step 6 To ensure that traffic processing has resumed, issue the command **rtrv-ne-health**.

Dynamic System Data

To modify the dynamic system data parameters in Table 3-4, use the sys_config_dynamic MML name variable for the **prov-add**, **prov-dlt**, and **prov-ed** commands. You need not halt and restart call processing for the changes to take effect.

In the following example, the **prov-add** command adds the dynamic system data parameter OVLDLEVEL1PERCENT to a dynamic configuration file. The **prov-ed** command modifies the value of the OVLDLEVEL1PERCENT parameter. The **prov-dlt** command deletes the OVLDLEVEL1PERCENT parameter from the dynamic configuration file.

Example

```
prov-add:name=sys_config_dynamic,OVLDLEVEL1PERCENT=20
prov-ed:name=sys_config_dynamic,OVLDLEVEL1PERCENT=25
prov-dlt:name=sys_config_dynamic,OVLDLEVEL1PERCENT
```

The MML commands write the parameters in Table 3-4 to a dynamic configuration file or to a section within a file.

Table 3-4 Dynamic System Data Parameters

Parameter	Description	Default
LOGDIRECTORY	Specifies the directory used when the active log file is created, and also specifies the directory where the rotated log file is stored.	/var/log/
LOGFILENAMEPREFIX	Specifies the filename prefix used when the log files are created or rotated. The .log postfix is appended to the end of the prefix to establish the name of the active log file.	platform.log
LOGPRIO	Defines the initial logging levels. By default it is set to TRACE. When the system initializes and is running, the levels set for individual packages (0x0000 to 0xFFFF) determine the log levels. See the "Logging Levels" section on page 4-10.	TRACE
LOGFILEROTATESIZE	Triggers a log file rotation based on the size of the active file. The application regularly checks the current size of the file to determine whether a rotation is required. If a file rotation is triggered by this parameter, the rotated file might be slightly larger than the size specified by this parameter. This parameter triggers a file rotation and also resets the timer associated with the LOGFILEROTATEINTERVAL parameter.	10 Mb
LOGFILEROTATEINTERVAL	Triggers a log file rotation based on the time elapsed since the previous rotation. This timer is reset after any rotation occurs, regardless of the cause or trigger of the rotation.	1440 minutes (24 hours)
IPADDRNMS	Defines the IP address of the network management system.	—
OVLDSAMPLERATE	Defines the frequency of CPU sampling and threshold checking.	3000 millisecond (ms) polling rate

Table 3-4 Dynamic System Data Parameters (continued)

Parameter	Description	Default
OVLDLEVEL1PERCENT	Indicates what percentage of calls should be rejected when an overload condition occurs. This parameter is used in conjunction with the OVLDLEVEL1FILTER parameter. The overload level 1 value is the lowest level of overload and must be less than or equal to the provisioned values for OVLDLEVEL2PERCENT and OVLDLEVEL3PERCENT. **Note** If this value is set to zero, no overload level 1 treatment occurs.	20
OVLDLEVEL1FILTER	Indicates what call types should be gapped if an overload level 1 condition occurs. The possible values are: • Normal—Emergency or priority calls are not gapped. • All—All calls are gapped, regardless of type. **Note** If the overload percentage is set to 100, all calls are gapped irrespective of this setting.	Normal
OVLDLEVEL1THRESHLOWER CALLS	Determines the number of active calls below which the application load must fall in order to remove the overload level 1 condition.	1800
OVLDLEVEL1THRESHUPPER CALLS	Determines how many simultaneous active calls trigger an overload level 1 condition.	1900
OVLDLEVEL1THRESHLOWER CPU	Determines the CPU utilization level below which the application must fall in order to remove the overload level 1 condition.	60
OVLDLEVEL1THRESHUPPER CPU	Determines the level of CPU utilization that triggers an overload level 1 condition.	65
OVLDLEVEL2PERCENT	Indicates what percentage of calls should be rejected when an overload condition occurs. The parameter is used in conjunction with the OVLDLEVEL2FILTER parameter. This is the second level of overload and must be less than or equal to the provisioned value of OVLDLEVEL3PERCENT and greater than or equal to the provisioned value of OVLDLEVEL1PERCENT. **Note** If this value is set to zero, no overload level 1 or 2 treatment occurs (by definition, the level 1 value must also be zero).	75
OVLDLEVEL2FILTER	Indicates what call types should be gapped if an overload level 2 condition occurs (see OVLDLEVEL1FILTER).	Normal
OVLDLEVEL2THRESHLOWER CALLS	Determines the number of active calls below which the application load must fall in order for the overload level 2 condition to be removed.	2000
OVLDLEVEL2THRESHUPPER CALLS	Determines how many simultaneous active calls trigger an overload level 2 condition.	2200
OVLDLEVEL2THRESHLOWER CPU	Determines the level of CPU utilization below which the application must fall in order for the overload level 2 condition to be removed.	70
OVLDLEVEL2THRESHUPPER CPU	Determines the level of CPU utilization that triggers an overload level 2 condition.	80

Table 3-4 Dynamic System Data Parameters (continued)

Parameter	Description	Default
OVLDLEVEL3PERCENT	Indicates what percentage of calls should be rejected when an overload condition occurs. The parameter is used in conjunction with the OVLDLEVEL3FILTER parameter. This is the highest level of overload and must be greater than or equal to the provisioned values for OVLDLEVEL1PERCENT and OVLDLEVEL2PERCENT. **Note** If this value is set to zero, no overload treatment occurs (by definition, the level 1 and level 2 values must also be zero).	90
OVLDLEVEL3FILTER	Indicates what call types should be gapped if an overload level 3 condition occurs (see OVLDLEVEL1FILTER).	Normal
OVLDLEVEL3THRESHLOWER CALLS	Determines the number of active calls below which the application load must fall in order to remove the overload level 3 condition.	2300
OVLDLEVEL3THRESHUPPER CALLS	Determines how many simultaneous active calls trigger an overload level 3 condition.	2400
OVLDLEVEL3THRESHLOWER CPU	Determines the level of CPU utilization below which the application must fall in order to remove the overload level 3 condition.	85
OVLDLEVEL3THRESHUPPER CPU	Determines the level of CPU utilization that triggers an overload level 3 condition.	95
CIAGENTSCANPERIOD	Specifies the frequency with which the CIagent polls the CPU utilization.	—
ALARMDEBOUNCETIME	Specifies the length of time that an alarm condition must persist before being reported, and any associated action taken.	0
CALLREFERENCEUSAGE	Determines which call reference identity is passed on to the PGW 2200 (call reference field or Conference ID).	—
DISKUSAGELIMIT	Represents a percentage of disk occupancy. The application continually polls the system for disk occupancy, and if the percentage rises above the limit set by DISKUSAGELIMIT, the LOW_DISK_SPACE alarm is raised. DISKUSAGELIMIT has a default value of 95 percent. The value range is 0–100, inclusive. When dynamically provisioned, the parameter DISKUSAGELIMIT, if not set within that range, is set to the default value (95) and the CONFIGURATION_ FAILURE alarm is raised.	95
RegFailureReleaseCause	This parameter specifies the Q.850 release cause, which the HSI uses after the HSI fails three times to register to a gatekeeper. This parameter is assigned a value in the range 1—127	—

H.323 Stack Configuration

The parameter name is based on the ASN.1 paths; but, in some cases, the parameter name has been shortened for convenience. For example, “capabilities” has been shortened to “caps.”

The case of the parameter name reflects exactly the ASN.1 definitions; but, case is not important to MML configuration.

Nonprovisionable Data

The parameters in Table 3-5 cannot be altered through MML commands.

Table 3-5 Nonprovisionable Data Parameters

H323_SYS	Description
system.manualstart	Present
system.pdlname	Absent
system.delimiter	#FF
ras.gatekeeper	Absent
ras.rasmulticastaddress	224.0.1.41.1718
h245.capabilities.manualoperation	Present
h245.masterslave.manualoperation	Present
q931.manualaccept	Present
q931.earlyH245	Present
q931.autoanswer	Present
q931.manualcallprocessing	Present
q931.h245tunneling	Present

MML Provisionable Data

H.323 System Parameters

The parameters in Table 3-6 are required for H.323 stack initialization. To modify the parameters in Table 3-6, use the h323_sys MML name variable for the **prov-add**, **prov-dlt,** and **prov-ed** commands. Stop and restart the application for these changes to take effect.

Note The asterisk (*) after a parameter name in the first column of Table 3-6 denotes a mandatory RADVision parameter that has an inbuilt default value if a value is not set in provisioning.

Table 3-6 H.323 System Initialization Parameters

Parameter	Description	Type	Example
maxCalls*	Maximum number of concurrent calls allowed	INTEGER(0, 65535)	2500
maxChannels*	Maximum number of concurrent channels allowed	INTEGER(0, 65535)	2

Q.931 Parameters

To modify the parameters listed in Table 3-7, use the q931 MML name variable for the **prov-add, prov-dlt,** and **prov-ed** commands.

In the following example, the **prov-add** command sets the Q.931 parameter maxCalls to the value 2000.

Example

```
prov-add:name=q931,maxCalls=2000
```

The Update Type column in Table 3-7 shows when the change to a parameter takes effect once a change is made:

- Immediate means that the effect of the change is immediate.
- Start means that the application needs to be restarted for the change to take effect.
- Next Call means that the next call has the new parameter set.

Note Immediate and Next Call update types refer to dynamic system data.

Note The asterisk (*) after a parameter name in the first column of Table 3-7 denotes a mandatory RADVision parameter with an inbuilt default value that will be used if the value is not set in provisioning.

Table 3-7 Q.931 Parameters

Parameter Name	Description	Type	Example	Update Type
responseTimeOut*	The maximum time (in seconds) permitted to receive the first response to a call. If this parameter expires, the call is disconnected.	INTEGER(1,200)	20	Immediate
connectTimeOut*	The maximum time (in seconds) the stack waits for call establishment after the first response is received. If this parameter expires, the call is disconnected.	INTEGER(1,20000)	180	Immediate
callSignalingPort*	The number of the port receiving the calls destined for the PGW 2200.	INTEGER(0,65535)	1720	Start
maxCalls*	The maximum number of simultaneous calls permitted. If this parameter is exceeded, the next call attempt returns busy.	INTEGER(0,65535)	2500	Next Call
notEstablishControl	The stack does not allow the switching of control from the Q.931 to the H.245 stack.	NULL	Not present	Next Call
overlappedSending	Because the Q.931 configuration flag indicates that both parties support overlap sending, this state notifies the other party that it can send an overlap sending message.	NULL	Present	Immediate

Note The Q.931 parameter overlappedSending has been combined with the RAS overlappedSending parameter. If you set the Q.931 overlappedSending parameter, you also set the RAS overlappedSending parameter.

RAS Parameters

The parameters in Table 3-8 are required for RAS stack initialization. To modify the RAS parameters, use the ras MML name variable for the **prov-add**, **prov-dlt,** and **prov-ed** commands.

In the following example, the **prov-add** command sets the RAS parameter maxfail to the value 3.

Example

```
prov-add:name=ras,maxfail=3
```

The array index [i] in some of the parameter names in the first column of Table 3-8 must be replaced with a valid braced index from 1 to 20, and must be continuous and unique (that is, it must contain no duplicates).

The Update Type column in Table 3-8 shows when the change to a parameter takes effect after it is modified:

- Immediate means that the effect of the change is immediate.
- Start means that the application needs to be restarted for the change to take effect.
- Next Call means that the next call has the new parameter set.

Note Immediate and next call update types are dynamic system data.

Note The RAS parameter overlappedSending is not available here because it has been combined with the Q.931 overlappedSending parameter. If you set the Q.931 overlappedSending parameter, you also set the the RAS overlappedSending parameter.

Note The asterisk (*) after a parameter name in the first column of Table 3-8 denotes a mandatory RADVision parameter with an inbuilt default value that will be used if the value is not set in provisioning.

Table 3-8 RAS Parameters

Parameter Name	Description	Type	Example	Update Type
manualRAS	If this parameter is present, the stack does not perform automatic RAS procedures (it waits to be driven by the application).	NULL	—	Start
responseTimeOut*	The time (in seconds) that the stack waits until it notifies the application that the called party has failed to respond to a transaction.	INTEGER(1, 200)	10	Immediate
maxFail*	Maximum number of retry gatekeeper registration attempts.	INTEGER(1, 200)	3	Immediate

Table 3-8 RAS Parameters (continued)

Parameter Name	Description	Type	Example	Update Type
allowCallsWhenNonReg	If this parameter is present, it allows calls to proceed even if gatekeeper registration has not been done for the PGW 2200.	NULL	Not present	Immediate
manualRegistration	If this parameter is present, the stack does not perform automatic gatekeeper registration procedures (it waits to be driven by the application).	NULL	Not present	Stop/Start
timeToLive	The maximum time (in seconds) that the registration of the PGW 2200 with a gatekeeper remains valid. The stack reregisters periodically.	INTEGER(1, 65535)	400	Immediate
rasPort*	The number of the port receiving all RAS transactions for the current endpoint. Set this parameter to 0 to allow the software to look for the available port.	INTEGER(0, 65535)	0	Start
compare15bitRasCrv	If this parameter is present, it causes the stack to ignore the call reference value (CRV) MSBit in RAS messages.	NULL	—	Immediate
maxRetries*	Maximum number of RAS retransmissions.	INTEGER(1, 200)	3	Immediate
maxMulticastTTL	Maximum number of multicast time to live (TTL).	INTEGER(0, 200)	3	Start
preGrantedArqUse	Choice of direct or routed. If direct, the pregranted Admission Request (ARQ) feature is used for both direct and routed calls. If routed, the pregranted ARQ feature is used only for routed calls. If absent, the pregranted ARQ is not used.	STRING	direct	Next Call
manualDiscovery.ipAddress	The IP address of a known gatekeeper with which an endpoint might attempt to register.	STRING	10.70.54.53	Start

Table 3-8 RAS Parameters (continued)

Parameter Name	Description	Type	Example	Update Type
manualDiscovery.port	The port associated with the manualDiscovery.ipAddress, which can, by agreement, be either a well-known port or another port.	INTEGER(0, 65535)	1719	Start
gateway.prefix[i]	The gateway registers the telephone prefix specified by this parameter to indicate that it is able to terminate it.	STRING	0208	Immediate
gatekeeperId	Identifies the gatekeeper with which the endpoint is trying to register.	STRING	OuterLondon	Immediate
terminalAlias[i].e164	Two variants of the same address for the endpoint; e164 is numeric and h323ID is text.	STRING	0208001000	Immediate
terminalAlias[i].h323ID		STRING	GW@ot.com.au	Immediate
endpointVendor.t35CountryCode	These parameters identify the manufacturer of the endpoint.	INTEGER(0, 255)	11	Immediate
endpointVendor.t35Extension		INTEGER(0, 255)	11	Immediate
endpointVendor.manufacturerCode		INTEGER(0, 65535)	9	Immediate
endpointVendor.productId	Data that the manufacturer assigns to each product.	STRING	H323ESP	Immediate
endpointVendor.versionId	Data that the manufacturer assigns to each version.	STRING	R0.2.4	Immediate

H.245 Parameters

To modify the H.245 parameters listed in Table 3-9, use the h245 MML name variable for the **prov-add, prov-dlt** and **prov-ed** commands.

In the following example, the **prov-add** command sets the H.245 parameter masterSlave.timeout to the value 5.

Example

```
prov-add:name=h245,masterSlave.timeout=5
```

The Update Type column in Table 3-9 shows when a change to an H.245 parameter takes effect after it is modified:

- Immediate means that the effect of the change is immediate.
- Start means that the application needs to be restarted for the change to take effect.
- Next Call means that the next call has the new parameter set.

Note Immediate and Next Call update types are dynamic system data.

Table 3-9 H.245 Parameters

Parameter Name	Description	Type	Example	Update Type
masterSlave.terminalType	The terminal type for the PGW 2200.	INTEGER(0, 255)	60	Next Call
masterSlave.manualResponse	If this parameter is present, it cancels automatic acknowledgment of master or slave determination.	NULL	Present	Next Call
masterSlave.timeout	The maximum time (in seconds) the stack waits before it gives up on the master/slave procedure.	INTEGER(0, 65535)	5	Immediate
channelsTimeout	The time (in seconds) the stack waits for a response to a channel establishment message.	INTEGER(0, 65535)	10	Immediate
roundTripTimeout	The time (in seconds) the stack waits for round-trip procedure completion.	INTEGER(0, 65535)	5	Immediate
requestCloseTimeout	The time (in seconds) the stack waits for request close procedure completion.	INTEGER(0, 65535)	5	Immediate
requestModeTimeout	The time (in seconds) the stack waits for request mode procedure completion.	INTEGER(0, 65535)	5	Immediate
caps.timeout	The maximum time (in seconds) the stack waits before it gives up on the capability exchange procedure.	INTEGER(0, 65535)	5	Immediate
caps.maxAudioDelay	Maximum H.255 multiplex audio delay jitter.	INTEGER(0, 1023)	60	Immediate
mediaLoopTimeout	The timeout (in seconds) of the media loop procedure.	INTEGER(0, 65535)	5	Immediate

Table 3-10, Table 3-11, and Table 3-12 list the parameters and modes related to the configuring of codecs. The array index [i] must be replaced with a valid braced index from 1 to 20. The braced index must be continuous and unique (that is, there must be no duplicates).

Table 3-10 H.245 Terminal Capability Codec Parameters

Parameter Name	Type
caps.table[i].entryNo	INTEGER(1, 65535)
caps.table[i].audio.g711Alaw64k	INTEGER(1, 256)
caps.table[i].audio.g711Alaw56k	INTEGER(1, 256)
caps.table[i].audio.g711Ulaw64k	INTEGER(1, 256)
caps.table[i].audio.g711Ulaw56k	INTEGER(1, 256)
caps.table[i].audio.g722at64k	INTEGER(1, 256)

Table 3-10 H.245 Terminal Capability Codec Parameters (continued)

Parameter Name	Type
caps.table[i].audio.g722at56k	INTEGER(1, 256)
caps.table[i].audio.g722at48k	INTEGER(1, 256)
caps.table[i].audio.g7231.maxAudioFrames	INTEGER(1,256)
caps.table[i].audio.g7231.silenceSuppression	INTEGER(0,1)
caps.table[i].audio.g728	INTEGER(1, 256)
caps.table[i].audio.g729	INTEGER(1, 256)

Table 3-11 H.245 Channel Codec Parameters

Parameter Name	Type
chan[i].name	STRING
chan[i].audio.g711Alaw64k	INTEGER(1, 256)
chan[i].audio.g711Alaw56k	INTEGER(1, 256)
chan[i].audio.g711Ulaw64k	INTEGER(1, 256)
chan[i].audio.g711Ulaw56k	INTEGER(1, 256)
chan[i].audio.g722at64k	INTEGER(1, 256)
chan[i].audio.g722at56k	INTEGER(1, 256)
chan[i].audio.g722at48k	INTEGER(1, 256)
chan[i].audio.g7231.maxAudioFrames	INTEGER(1,256)
chan[i].audio.g7231.silenceSuppression	INTEGER(0,1)
chan[i].audio.g728	INTEGER(1, 256)
chan[i].audio.g729	INTEGER(1, 256)

Table 3-12 H.245 Modes

Parameter Name	Type
modes[i].name	STRING
modes[i].audio.g711Alaw64k	NULL
modes[i].audio.g711Alaw56k	NULL
modes[i].audio.g711Ulaw64k	NULL
modes[i].audio.g711Ulaw56k	NULL
modes[i].audio.g722at64k	NULL
modes[i].audio.g722at56k	NULL
modes[i].audio.g722at48k	NULL
modes[i].audio.g7231	INTEGER(1,256)
modes[i].audio.g728	NULL
modes[i].audio.g729	NULL

Codec Selection

The Cisco HSI negotiates the media stream codec to establish a match between the PSTN MGCP media gateway (for example, the Cisco AS5xxx series or Cisco MGX series) and the H.323 endpoint or gateway. To match codecs, the MGCP gateway must be configured to match what is expected at the H.323 end. Similarly, the Cisco HSI also must be configured with the same codecs.

The Cisco HSI receives a list of codecs from the MGCP gateway and matches the listed codecs to the codecs that are configured on the HSI. The HSI advertises all of the successful matches in the H.245 terminalCapabilitySet messaging with the H.323 endpoint.

It is important to determine and configure the “frames-per-packet” value correctly on the Cisco HSI per codec. If “frames-per-packet” value is incorrect, the codec may not be negotiated successfully between the HSI and the H.323 endpoint.

It is also important to configure the MGCP gateway correctly. The gateway should be configured to provide “static payload” values for the required codecs, rather than dynamic payload types (see Table 4 in RFC 3551, Schulzrinne and Casner).

Quick Reference for Important Parameters

Table 3-13, Table 3-14, Table 3-15, and Table 3-16 can be used in initial HSI configuration. The tables present parameters that you might use frequently to align the Cisco HSI with an existing PSTN or Voice over IP network.

Table 3-13 presents important call control parameters.

Table 3-13 Common Call Control Parameters

Parameter Name	Parameter Value	Description
A_CC_oLinecall	0—Unknown 10—Ordinary	Calling party's category
A_CC_Clir	0—No indication 1—Presentation allowed 2—Presentation restricted 3—Address not available	Address presentation restricted indicator
A_CC_ANumDataSI	0—None 1—User provided not verified 2—User provided verified passed 3—User provided verified failed 4—Network provided	Screening indicator
A_CC_oIsdnAllTheWay	0—ISDN user part not used all the way 1—ISDN user part used all the way	Forward call indicator, ISUP indicator
A_CC_oIsdnPref	0—ISDN user part preferred all the way 1—ISDN user part not required all the way 2—ISDN user part required all the way	Forward call indicator, ISUP preference

Table 3-13 Common Call Control Parameters (continued)

Parameter Name	Parameter Value	Description
A_CC_Interworking	0—No interworking encountered (SS7 all the way) 1—Interworking encountered	Backward call indicator, Interworking indicator
A_CC_Location	1—User 2—Private local 3—Public local 4—Transit 5—Public remote 6—Private remote 7—International 8—Interworking 9—Local interface 11—Local remote 12—Packet manager 13—Unknown	Cause indicator, Location

The following MML command example shows the command sequence used to provision the call control parameters provided in the preceding table.

Example

```
mml
> prov-sta::srcver=active, dstver=myconf
> prov-ed:name=CCPackage, A_CC_ANumDataSI=2
> prov-cpy
> restart-softw
```

Table 3-14 presents important static system data parameters.

Table 3-14 Common Static System Data Parameters

Parameter Name	Parameter Values	Description
CarrierCodeMapping	• "enabled"—a string that indicates the feature is enabled. • Blank ("")—indicates the feature is disabled. • "deleted"—indicates that the feature is disabled.	Allows the mapping of a special tech prefix (the format of which is *CCxCy*) to the DestinationCircuitID "group" field in the ARQ message. This feature works only with IOS Gatekeeper build Release 12.2(15)T10 or above.
ClipClirSupported	• "enabled"—a string that indicates the feature is enabled. • Blank ("")—indicates the feature is disabled • "deleted"—indicates that the feature is disabled	Allows transit of CLI presentation/screening information. **Note** Setting this parameter to "enabled" enables use of Caller ID.
DtmfSupportedType	• "dtmf"—the recommended value for interworking with Cisco gateways • "basicString"	Selects the DTMF type during H.245 terminal capabilities exchange. **Note** Set this parameter to "dtmf" and the DtmfSupportedDirection parameter to "both" to enable DTMF support.
DtmfSupportedDirection	• "tx"—transmit to H323 endpoint • "rx"—receive from H.323 endpoint • "both"—transmit and receive DTMF • Blank (""), "deleted," or any other string, such as "disabled"—indicates the feature is disabled	Selects DTMF transit direction. **Note** Set this parameter to "both" and the DtmfSupportedType parameter to "dtmf" to enable DTMF support.
H225PavoSupported	• "enabled"—a string that indicates the feature is enabled. • Blank ("")—indicates the feature is disabled • "deleted"—indicates that the feature is disabled	Allows transit of redirecting number parameter (contained in Cisco CallManager H.225 setup messages—nonStandardControl field).
RaiSupported	For example: • "enabled"—a string that indicates the feature is enabled. • Blank ("")—indicates the feature is disabled • "deleted"—indicates that the feature is disabled	Allows H.225 RAS RAI messages to be sent to the gatekeeper if the E-ISUP link fails or if the HSI is under heavy load. **Note** Set this parameter to "enabled" to enable the HSI to support RAI messages.

Table 3-14 Common Static System Data Parameters (continued)

Parameter Name	Parameter Values	Description
NotifyMsgEnabled	For example: • "enabled"—a string that indicates the feature is enabled. • Blank ("")—indicates the feature is disabled • "deleted"—indicates that the feature is disabled	Allows transit of connected number, display information, and generic notification indicator in H.225 Notify messages.
VSCB_IPADDR1/2	IP address, for example: "10.10.10.1"	Allows IP address configuration of second PGW.
VSCB_PORT_NUMBER1/2	Port number, for example: 8003	Allows port configuration of second PGW.

The following MML command example shows the command sequence used to provision the static system data parameters provided in the preceding table.

Example

```
mml
> prov-sta::srcver=active, dstver=myconf
> prov-ed:name=SYS_CONFIG_STATIC, DtmfSupportedType="dtmf"
> prov-cpy
> restart-softw
```

Table 3-15 presents common RAS parameters.

Table 3-15 Common RAS Parameters

Parameter Name	Parameter Value	Description
gateway.prefix[1] gateway.prefix[2]	For example: 020	HSI prefix (for gatekeeper registration)
timeToLive	Integer (to specify number of seconds) for example, 45 **Note** To enable lightweight RRQs, the value for this parameter should be set substantially lower than the default (600).	RAS registration time to live. See Table 3-8.

The following MML command example shows the command sequence used to provision the RAS parameters provided in the preceding table.

Example

```
mml
> prov-sta::srcver=active, dstver=myconf
> prov-ed:name=RAS, timeToLive=45
> prov-cpy
> restart-softw
```

Table 3-16 presents common H.245 parameters for enabling the G.729 codec.

Table 3-16 Common H.245 Parameters

Parameter Name	Parameter Value
chan[i].name	For example: prov-add:name=”H245”,chan[4].name=”g729”
chan[i].audio.g729	For example: prov-add:name=”H245”,chan[4].audio.g729=”2”
caps.table[i].audio.g729	For example: prov-add:name=”H245”,caps.table[4].audio.g729=”2”
caps.table[i].entryNo	For example: prov-add:name=”H245”,caps.table[4].entryno=”729”
modes[i].name	For example: prov-add:name=”H245”,modes[3].name=”g729”
modes[i].audio.g729	For example: prov-add:name=”H245”,modes[3].audio.g729=”3].audio.g729=””

The following MML command example shows the command sequence used to provision the H.245 parameters provided in the preceding table for enabling the G.729 codec. Provisioning the G.729 codec on the Cisco HSI supports passing SS7 calls to the Cisco CallManager through a gateway running the Media Gateway Control Protocol (MGCP).

Example

```
prov-sta::srcver=”active”,dstver=”g729"
prov-add:name=”H245”,caps.table[4].audio.g729=”2”
prov-add:name=”H245”,caps.table[4].entryno=”729”
prov-add:name=”H245”,chan[4].audio.g729=”2”
prov-add:name=”H245”,chan[4].name=”g729”
prov-add:name="H245",modes[3].audio.g729=""
```
prov-add:name=”H245”,modes[3].name=”g729”

HSI Feature Configuration

This section describes how to enable the following HSI features:

- Asymmetric Codec Treatment
- Empty Capability Set
- H.323 Hairpin
- T.38 Fax
- HSI INFORMATION Message Support
- HSI Support for Tech Prefixes
- Configuring Clear Channel on the Cisco HSI

- Configuring G.726 on the Cisco HSI
- Configuring G.729 Annex and G.729 Annex B

Asymmetric Codec Treatment

The Asymmetric Codec Treatment feature averts the potential for inconsistencies in codec selection, which can result if the open channel requests are sent by each endpoint at nearly the same time, so that neither side has received an open channel request prior to sending one. In practice, such asymmetric conditions occur only for slow start calls. When there is a fast start recipient, both channels agree to use the same codec in unison.

The Asymmetric Codec Treatment support is enabled if this parameter is present and set to anything other than "". For example, support is enabled if the parameter is explicitly set to "Enabled." To enable Asymmetric Codec Treatment, enter the following command:

Example:

```
prov-add:name=sys_config_static, asymmetrichandlingsupported = "Enabled"
```

Empty Capability Set

The Empty Capability Set feature enables an H.323 endpoint to send a TCS message with empty capabilities during a call. The TCS message causes the audio channels to close. This action enables the negotiation and opening of new audio channels.

The Empty Capability Set feature is useful when the H.323 endpoint wishes to change the audio codec during a call or if the endpoint needs to divert the media streams to a different location. Typically, the feature is used to place a call on hold to disable the media stream until the user presses the Resume button.

The Empty Capability Set feature on the HSI requires no provisioning.

H.323 Hairpin

The H.323 Hairpin feature can be used to connect a call between two H.323 endpoints without using resources on the media gateway. For example, the PGW can respond to the dialled number in an incoming H.323 call by routing the call to another HSI (perhaps the same HSI) rather than routing the call to the PSTN. In this case, the originating and terminating HSIs establish the call normally but pass the H.245 address of the H.323 endpoints. This enables the two endpoints to use H.245 to negotiate media channels with each other directly, independent of the HSI.

The H.323 Hairpin feature on the HSI requires no provisioning. However, to operate throughout the system, H.323 Hairpin must be enabled on the PGW. On the PGW, you enable H.323 Hairpin through a trunk group property by issuing the following commands:

```
prov-add:trnkgrpprop:name="2000",AllowH323Hairpin="1"
prov-add:trnkgrpprop:name="3000",AllowH323Hairpin="1"
```

Note H.323 Hairpin must be enabled for both the ingress and egress EISUP trunk groups.

Refer to Cisco PGW and Cisco IOS documentation at www.cisco.com for further information on these commands.

T.38 Fax

The T.38 Fax feature enables the HSI to alter a call, initially established for voice, to support a fax transmission.

When a fax call is initiated, a voice call is established. When the terminating gateway detects the fax tone generated by the terminating fax machine, the gateway initiates a T.38 mode request using H.245 procedures from the terminating gateway. If the opposite end of the call acknowledges the T.38 mode request, the initial audio channel is closed and a T.38 fax relay channel is opened.

You enable T.38 Fax for the HSI by specifying static system data parameters. By default, T.38 is provisioned on the HSI by use of the following commands:

```
prov-add:name=sys_config_static,t38maxval="MaxBit 0x90, FxMaxBuf 0xc8, FxMaxData 0x48"
prov-add:name=sys_config_static,t38options="FxFillBit 0, FxTransMMR 0, FxTransJBIG 0,
FxRate Trans, FxUdpEC Red"
```

Table 3-3 describes the T.38 static system data parameters. The T.38 parameters for HSI correspond to T.38 parameters proposed in the ITU T.38 recommendation.

Configuring T.38 Fax on the Cisco PSTN Gateway

To enable T.38 Fax throughout the system, you must enable T.38 Fax on the Cisco PGW. On the PGW, T.38 is enabled through a trunk group property by use of the following MML command:

```
prov-add:trnkgrpprop:name="2000",FaxSupport="1"
```

Configuring T.38 Fax on a Cisco IOS H.323 Gateway

Enable T.38 Fax on a Cisco IOS H.323 gateway by issuing the following IOS commands:

```
voice service voip
     fax protocol t38 ls-redundancy 0 hs-redundancy 0 fallback none
```

Configuring T.38 Fax on a Cisco IOS MGCP Gateway

Enable T.38 fax on a Cisco IOS MGCP gateway by issuing the following IOS commands:

```
voice service voip
     fax protocol t38 ls-redundancy 0 hs-redundancy 0 fallback none
       mgcp package-capability fxr-package
```

Refer to PGW and Cisco IOS documentation at www.cisco.com for further information on these commands.

HSI INFORMATION Message Support

Cisco CallManager uses the H.225 INFORMATION message during transfer to indicate that ringback tone is on or off. The Cisco HSI now supports this message to correctly interoperate with Cisco CallManager.

Support for the H.225 INFORMATION message is enabled by default. A crafts person can disable H.255 INFORMATION message support through a new property called Information MsgDisabled by issuing the following MML command:

```
prov-add:name=sys_config_static,informationmsgdisabled = "True"
```

HSI Support for Tech Prefixes

The Cisco HSI now maps the '*' (asterisk, or star) and '#' (number sign, or hash) H.225 prefixes to the PGW for H.323 to PSTN calls as follows:

- '*' to the value provisioned in ccpackage.Star
- '#' to the value provisioned in ccpackage.Hash
- The current value for ccpackage.Star is 'B'.
- The current value for ccpackage.Hash is 'A'.

The crafts person can change these values by issuing the following MML command:

```
prov-ed:name=ccpackage,hash='C'
```

Cisco HSI now maps the EISUP 'B' to '*' and 'C' to '#' (Called Party Number) for PSTN to H.323 calls.

Configuring Clear Channel on the Cisco HSI

The Clear Channel capability (identified as G.Clear or gclear in this document) enables support for both voice and data calls on a network. However, the end applications are responsible for packet loss and error recovery. For more information, refer to the document *G.Clear, GSMFR, and G.726 Codecs and Modem and Fax Passthrough for Cisco Universal Gateways* at http://www.cisco.com/en/US/products/sw/iosswrel/ps1839/products_feature_guide09186a00800b3568.html.

Note In association with the Cisco HSI, the Cisco PGW must be running 9.5(2) patch set gs034/nn028, or later, to use G.Clear.

The Cisco HSI interoperates with Cisco voice gateways (for example, the Cisco AS54xx series or VISM), which advertises G.Clear capability via MGCP signaling using the following methods: G.Clear, G.nX64, CCD. The Cisco HSI automatically selects the correct method depending on the gateway that originates or terminates the call.

Refer to the *Cisco H.323 Signaling Interface User Guide* for information regarding the use of HSI MML commands.

Table 3-17 presents examples of configuration commands that may be required to implement a particular G.Clear configuration.

Table 3-17 Configuring Clear Channel

Clear Channel Parameters	Example Value	Example Configuration
H245, caps.table[i].audio.gclear	"ClearChid" **Note** The string "ClearChid" is case-sensitive; it must be entered exactly as displayed in all command examples in this table.	prov-add:name=h245, caps.table[9].audio.gclear="ClearChid" prov-add:name=h245, caps.table[10].audio.gclear="ClearChid"
H245, caps.table[i].audio.entryNo	1010, 1011, 1012… **Note** This parameter should be set to a unique integer value.	prov-add:name=h245, caps.table[9].entryNo=1010 prov-add:name=h245, caps.table[10].entryNo=1011
H245, chan[i].audio.gclear	"ClearChid"	prov-add:name=h245, chan[9].audio.gclear=ClearChid" prov-add:name=h245, chan[10].audio.gclear="ClearChid"
H245, chan[i].name	"ClearChid"	prov-add:name=h245, chan[9].name="ClearChid" prov-add:name=h245, chan[10].name="ClearChid"
H245, modes[i].audio.gclear	"ClearChid"	prov-add:name=h245, modes[9].audio.gclear="ClearChid" prov-add:name=h245, modes[10].audio.gclear="ClearChid"
H245, modes[i].name	"ClearChid"	prov-add:name=h245, modes[9].name="ClearChid" prov-add:name=h245, modes[10].name="ClearChid"

Configuring G.726 on the Cisco HSI

The G.726 codec enables transcoding a PCM channel to or from an ADPCM data stream. The standard supports four data rates:16, 24, 32 and 40 kbit/sec.

G.726 capability is advertised by the Cisco HSI and other H.323 gateways/endpoints in H.225 fast-start elements, in H.245 (tunneled or a separate TCP/IP connection) terminal capability (TCS) messages, and open logical channel (OLC) messages.

Currently, H.323 devices use several different methods to advertise G.726. ITU G.726 Annex B defines one method, referred to in this document as g726-generic. Cisco H.323 gateways (for example, the Cisco AS5400) support an alternate method referred to as g726-cisco. There is another method used by the OpenH323 project; however, the Cisco HSI does not support that method.

MGCP gateways advertise G.726 capability using the method described in RFC 3551 (RTP Profile for Audio and Video Conferences with Minimal Control). The four data rates use dynamic payloads; however, the 32kbit/sec data rate, alternatively, can have a static payload value of 2 (this alternative value is being phased out).

You can configure the Cisco HSI for 32kbit/sec MGCP support using dynamic or static payload values. In addition, you can configure the Cisco HSI to support g726-generic and/or g726-cisco for the H.323 signaling. If possible, it is best to select g726-cisco for your network because it offers additional flexibility.

The g726-generic method cannot indicate the data rate in H.245 TCS messages. The ITU standard specifies that the data rate is only advertised in the OLC messages.

Note The H.245 ASN.1 syntax supports advertising the bitrate in TCS messages; however, G.726 Annex B prohibits advertising the bitrate in TCS messages. The Cisco HSI advertises the bitrate in the TCS messages as a “hint”; however, H.323 gateways/endpoints might not extract the field and take advantage of the presence of the bitrate in the TCS message.

The fact that the g726-generic method cannot indicate the data rate in an H.245 TCS message is not a problem if the MGCP gateway and your network are designed to support all data rates for this codec. However, if all data rates are not supported, it is possible for the remote endpoint/gateway to select a non-preferred or non-supported data rate in the OLC message.

Note For example, a data-rate preference list may establish the following order: G.726-16kbit/sec (highest preference), G.711-Alaw (second preference), G.726-24kbit/sec (lowest preference). In this case, a remote endpoint could select G.726-24kbit/sec in the OLC message; whereas, the Cisco HSI would prefer G.726-16kbit/sec. In this example, the next preferred codec ought to be G.711 A-law and not G.726-24kbit/sec. However, the g726-generic limitation enables the remote endpoint to select the least preferred codec.

If a data-rate preference list specifies only a single rate (for example, G.726-16kbit/sec), it is not possible to advertise this fact in the TCS message. Subsequently, the remote endpoint may attempt to open the media stream using an unsupported data rate (perhaps, G.726-24kbit/sec).

Whenever OLC messages are exchanged and a non-supported G.726 data rate is detected, to prevent unnecessary call clearing, the Cisco HSI always attempts to send the data rate selection to the MGCP gateway. If the MGCP gateway does not support the selected data rate, it sends a message to the Cisco PGW to clear the call.

If a non-preferred G.726 data rate is selected over a higher-preference codec, the HSI will continue with the call using the non-preferred data rate. This is preferable to the alternative (aborting the media stream, invoking an empty capability exchange followed by a re-negotiation of codecs and new OLC messaging). The alternative causes call processing delay and overhead associated with switching media streams.

Note The g726-cisco method avoids impaired or delayed processing because it advertises the data rate in the TCS messaging.

Refer to the *Cisco H.323 Signaling Interface User Guide* for information about Cisco HSI MML commands.

Table 3-18 presents examples of configuration commands that may be required to implement a particular G.726 configuration.

Table 3-18 Configuring G.726

G.726 Parameter	Example Value	Configuration Example
Configuring the Payload Type for the MGCP		
sys_config_static, UseG726StaticPayload	“enabled”, “true”, “” **Note** If this parameter is set to any text value, the Cisco HSI uses static payload value '2' to represent G.726 32kbit/sec to the MGCP gateway. If the parameter is deleted or is set to an empty string (“”), the HSI uses the default, dynamic-payload behavior.	prov-add:name=sys_config_static, UseG726StaticPayload="enabled" prov-ed:name=sys_config_static, UseG726StaticPayload=""
Configuring Cisco HSI g726-cisco		
H245, caps.table[i].audio.g726-cisco	“G726r16”, “G726r24”, “G726r32”, “G726r40” **Note** These string values are case-sensitive, and must be entered exactly as displayed in the commands in this table.	prov-add:name=h245, caps.table[5].audio.g726-cisco="G726r16" prov-add:name=h245, caps.table[6].audio.g726-cisco="G726r24"

Table 3-18 Configuring G.726 (continued)

G.726 Parameter	Example Value	Configuration Example
H245, caps.table[i].entryNo	7261, 7262, … **Note** Set this parameter to a unique integer value	prov-add:name=h245, caps.table[5].entryNo=7261 prov-add:name=h245, caps.table[6].entryNo=7262
H245, chan[i].audio.g726-cisco	“G726r16” “G726r24” “G726r32” “G726r40”	prov-add:name=h245, chan[5].audio.g726-cisco="G726r16" prov-add:name=h245, chan[6].audio.g726-cisco="G726r24"
H245, chan[i].name	“G726r16” “G726r24” “G726r32” “G726r40”	prov-add:name=h245, chan[5].name="G726r16" prov-add:name=h245, chan[6].name="G726r24"
H245, chan[i].audio.g726-cisco	“G726r16” “G726r24” “G726r32” “G726r40”	prov-add:name=h245, chan[5].audio.g726-cisco="G726r16" prov-add:name=h245, chan[6].audio.g726-cisco="G726r24"
H245, modes[i].audio.g726-cisco	“G726r16” “G726r24” “G726r32” “G726r40”	prov-add:name=h245, modes[5].audio.g726-cisco="G726r16" prov-add:name=h245, modes[6].audio.g726-cisco="G726r24"
H245, modes[i].name	“G726r16” “G726r24” “G726r32” “G726r40”	prov-add:name=h245, modes[5].name="G726r16" prov-add:name=h245, modes[6].name="G726r24"
Configuring Cisco HSI g726-generic		
H245, caps.table[i].audio.g726-generic	“generic”	prov-add:name=h245, caps.table[7].audio.g726-generic="generic" prov-add:name=h245, caps.table[8].audio.g726-generic="generic"

Table 3-18 Configuring G.726 (continued)

G.726 Parameter	Example Value	Configuration Example
H245, caps.table[i].audio.g726-generic.bitOrder	1,2 or 3 **Note** This field is a bitmask of 8 bits, and can take any value from 0...255. Refer to G.726 Annex B, section B4.2 for a more detailed description. The value in this field must match the value advertised by the H.323 endpoint/ gateways.	prov-add:name=h245, caps.table[7].audio.g726-generic.bitOrder=2 prov-add:name=h245, caps.table[8].audio.g726-generic.bitOrder=3
H245, caps.table[i].audio.g726-generic.maxSPP	30, 40 **Note** This field is an integer value from 0...65535.	prov-add:name=h245, caps.table[7].audio.g726-generic.maxSPP=30 prov-add:name=h245, caps.table[8].audio.g726-generic.maxSPP=40
H245, caps.table[i].entryNo	7263, 7264 **Note** Set this parameter to a unique integer value.	prov-add:name=h245, caps.table[7].entryNo=7263 prov-add:name=h245, caps.table[8].entryNo=7264
H245, chan[i].audio.g726-generic	“generic”	prov-add:name=h245, chan[7].audio.g726-generic="generic" prov-add:name=h245, chan[8].audio.g726-generic="generic"
H245, chan[i].audio.g726-generic.bitOrder	1,2 or 3	prov-add:name=h245, caps.table[7].audio.g726-generic.bitOrder=2 prov-add:name=h245, caps.table[8].audio.g726-generic.bitOrder=3
H245, chan[i].audio.g726-generic.maxSPP	30, 40	prov-add:name=h245, chan[7].audio.g726-generic.maxSPP=30 prov-add:name=h245, chan[8].audio.g726-generic.maxSPP=40
H245, chan[i].name	“g726-generic-16” “g726-generic-24” “g726-generic-32” “g726-generic-40”	prov-add:name=h245, chan[7].name="g726-generic-16" prov-add:name=h245, chan[8].name="g726-generic-24"
H245, modes[i].audio.g726-generic	“generic”	prov-add:name=h245, modes[7].audio.g726-generic="generic" prov-add:name=h245, modes[8].audio.g726-generic="generic"

Table 3-18 Configuring G.726 (continued)

G.726 Parameter	Example Value	Configuration Example
H245, modes[i].audio.g726-generic.bitOrder	1, 2 or 3	prov-add:name=h245, modes.table[7].audio.g726-generic.bitOrder=2 prov-add:name=h245, modes.table[8].audio.g726-generic.bitOrder=3
H245, modes[i].audio.g726-generic.maxSPP	30, 40	prov-add:name=h245, modes[7].audio.g726-generic.maxSPP=30 prov-add:name=h245, modes[8].audio.g726-generic.maxSPP=40
H245, modes[i].name	“g726-generic-16” “g726-generic-24” “g726-generic-32” “g726-generic-40”	prov-add:name=h245, modes[7].name="g726-generic-16" prov-add:name=h245, modes[8].name="g726-generic-24"

Configuring G.729 Annex and G.729 Annex B

Table 3-18 presents examples of configuration commands that may be required to implement a particular configuration of G.729 Annex A or G.729 Annex B.

Table 3-19 Configuring G.729 Annex A and G.729 Annex B

G.729 Parameter	Example Value	Example Configuration
H245,caps.table[i].audio.g729AnnexA	2, 3	prov-add:name=h245, caps.table[4].audio.g729AnnexA=2 prov-add:name=h245, caps.table[5].audio.g729AnnexB=3 prov-add:name=h245 caps.table[6].audio.g729AnnexAwAnnexB=2
H245,caps.table[i].entryNo	7290, 7291, 7292	prov-add:name=h245, caps.table[4].entryno=7290 prov-add:name=h245, caps.table[5].entryno=7291 prov-add:name=h245, caps.table[6].entryno=7292
H245,chan[i].name	“g729AnnexA” “g729AnnexB” “g729AnnexAwAnnexB”	prov-add:name=h245, chan[4].name="g729AnnexA" prov-add:name=h245, chan[5].name="g729AnnexB" prov-add:name=h245, chan[6].name="g729AnnexAwAnnexB"
H245,chan[i].audio.g729AnnexA	2, 3	prov-add:name=h245, chan[4].audio.g729AnnexA=2 prov-add:name=h245, chan[5].audio.g729AnnexB=3 prov-add:name=h245, chan[6].audio.g729AnnexAwAnnexB=2

Table 3-19 Configuring G.729 Annex A and G.729 Annex B (continued)

G.729 Parameter	Example Value	Example Configuration
H245,modes[i].name	“g729AnnexA”	prov-add:name=h245,modes[4].name="g729AnnexA"
	“g729AnnexB”	prov-add:name=h245,modes[5].name="g729AnnexB"
	“g729AnnexAwAnnexB”	prov-add:name=h245,modes[6].name="g729AnnexAwAnnexB"
H245,modes[i].audio.g729AnnexA	“”	prov-add:name=h245, modes[4].audio.g729AnnexA=""
		prov-add:name=h245, modes[5].audio.g729AnnexB=""
		prov-add:name=h245, modes[6].audio.g729AnnexAwAnnexB=""

Managing the Cisco HSI

Introduction

This chapter provides information about operation and management tasks for the Cisco H.323 Signaling Interface (HSI) application. This chapter contains the following sections:

Restarting the Cisco HSI Application

To restart the Cisco HSI at the MML command prompt, use the **restart-softw** MML command. For more information about this command, see Appendix A, "MML User Interface and Command Reference."

To start the Cisco HSI application, see the "Installing the Cisco HSI" section on page 2-2.

Stopping Call Processing

To stop call processing, use the **stp-callproc** MML command. This command causes the handling of new call requests to cease immediately, and, if no timeout period is specified, all existing calls are released immediately. If a timeout period is specified, existing calls are released after the specified amount of time has elapsed. For more information about the **stp-callproc** command, see Appendix A, "MML User Interface and Command Reference."

Starting Call Processing

To start call processing, use the **sta-callproc** MML command. For more information about this command, see Appendix A, "MML User Interface and Command Reference."

Stopping the Call Processing Application

To stop the call processing application, use the **stp-softw** MML command. For more information about this command, see Appendix A, "MML User Interface and Command Reference."

Starting the Call Processing Application

To start the call processing application, use the **sta-softw** MML command. For more information about this command, see Appendix A, "MML User Interface and Command Reference."

Reporting the Cisco HSI Status

To display the status of the Cisco HSI, use the **rtrv-softw** MML command. For more information about this command, see Appendix A, "MML User Interface and Command Reference."

Measurements

The following sections describe two measurement categories:

- System-related measurements
- Call-related measurements

System-Related Measurements

The CIagent is a Simple Network Management Protocol (SNMP) subagent. It handles the collection and storage of the following system performance measurements:

- CPU occupancy
- RAM occupancy
- Disk occupancy
- TCP usage

Use the CIAGENTSCANPERIOD parameter to define the period that the CIagent polls the CPU for utilization (see Chapter 3, "Provisioning the Cisco HSI").

Call-Related Measurements

The Cisco HSI application handles all call-related measurements. An SNMP MIB handles the collection of call-related measurement data.

The call-related measurements are organized into counter groups. The following MML counter groups are required:

- RAS (see Table 4-1 on page 4-3)
- Q.931 (see Table 4-2 on page 4-4)
- H.245 (see Table 4-3 on page 4-5)

The measurements in these groups are written to a file on disk every 30 minutes. The file name includes the date and time that measurements were written to disk.

Table 4-1 RAS Counter Group

Counter Name	Measurement	Type	Comments
GK_DISC_ATT_TOT	Gatekeeper discovery attempts	Integer	Incremented for every unicast gatekeeper request (GRQ) sent or for every multicast operation
GK_REG_ATT_TOT	Registration request attempts	Integer	Incremented for every registration request (RRQ) sent
GK_REG_SUCC_TOT	Registration request successes	Integer	Incremented for every registration confirmation (RCF) received
GK_RCV_UNR_ATT_TOT	GK- initiated unregistration attempts	Integer	Incremented for every unregistration request (URQ) received from a gatekeeper (GK)
GK_XMIT_UNR_SUCC_TOT	GK-initiated unregistration successes	Integer	Incremented for every unregistration confirmation (UCF) sent to a GK
GK_XMIT_UNR_ATT_TOT	T- initiated unregistration attempts	Integer	Incremented for every URQ sent to a GK
GK_RCV_UNR_SUCC_TOT	T- initiated unregistration successes	Integer	Incremented for every UCF received from a GK
GK_RLS_ATT_TOT	Disengage attempts	Integer	Incremented for every disengage request (DRQ) sent to a GK
GK_RLS_SUCC_TOT	Disengage successes	Integer	Incremented for every disengage confirmation (DCF) returned by a GK
GK_INFO_REPORT_TOT	Information reports	Integer	Incremented for every information request (IRQ) sent to the GK

Table 4-2 Q.931 Counter Group

Counter Name	Measurement	Type	Comments
FC_INC_CALL_ATT_TOT	H.225 Incoming Fast Connect Call Attempts	Integer	Incremented when a setup containing the fastStart element is received.
FC_INC_CALL_SUCC_TOT	H.225 Incoming Fast Connect Call Successes	Integer	Incremented when the Fast Connect procedure is used to establish an incoming H.323 call.
FC_OTG _CALL_ATT_TOT	H.225 Outgoing Fast Connect Call Attempts	Integer	Incremented when a setup containing the fastStart element is sent to an H.323 endpoint. Decremented when you revert to Version 1 signaling (another measurement incremented).
FC_OTG_CALL_SUCC_TOT	H.225 Outgoing Fast Connect Call Successes	Integer	Incremented when the Fast Connect procedure is used to establish an outgoing H.323 call.
V1_INC_CALL_ATT_TOT	H.225 Incoming Version 1 Call Attempts	Integer	Incremented when an incoming H.323 Version 1 Setup is received (that is, no fastStart element or H.245 tunneling).
V1_INC_CALL_SUCC_TOT	H.225 Incoming Version 1 Call Successes	Integer	Incremented when an incoming H.323 Version 1 call is established.
V1_OTG_CALL_ATT_TOT	H.225 Outgoing Version 1 Call Attempts	Integer	Incremented when an outgoing H.323 call reverts to Version 1 signaling.
V1_OTG_CALL_SUCC_TOT	H.225 Outgoing Version 1 Call Successes	Integer	Incremented when an outgoing H.323 call using Version 1 is established.
INC_NORM_REL_TOT	H.225 Incoming Call Normal Releases	Integer	Incremented when an established incoming H.323 call is taken down due to user on-hook.
INC_ABNORM_REL_TOT	H.225 Incoming Call Abnormal Releases	Integer	Incremented when an established incoming H.323 call is taken down due to anything other than user on-hook.
OTG_NORM_REL_TOT	H.225 Outgoing Call Normal Releases	Integer	Incremented when an established outgoing H.323 call is taken down due to user on-hook.
OTG_ABNORM_REL_TOT	H.225 Outgoing Call Abnormal Releases	Integer	Incremented when an established outgoing H.323 call is taken down due to anything other than user on-hook.
PGW_T38_FAX_ATT_TOT	Q931	Integer	Incremented for each T.38 Fax Call request from the PGW. Collection Intervals are provisionable (default is 12 hours).
PGW_T38_FAX_SUCC_TOT	Q931	Integer	Incremented for each T.38 Fax Call request from the PGW that is successfully reconfigured for T.38. Collection Intervals: Provisionable (default 12 hours)
H323_INTERWORK_SUCC_	Q931	Integer	Incremented for each successful H.323-H.323 interworking condition. Collection Intervals are provisionable (default is 12 hours).

Table 4-3 H.245 Counter Group

Counter Name	Measurement	Type	Comments
MASTER_SLAVE_ATT_TOT	H.245 Master Slave Determination Attempts	Integer	Incremented whenever either side of the call initiates the master slave determination procedure (using either H.245 tunneling or a separate H.245 signaling path).
MASTER_SLAVE_SUCC_TOT	H.245 Master Slave Determination Successes	Integer	Incremented whenever a master slave determination procedure is completed.
TERM_CAP_XCHG_ATT_TOT	H.245 Terminal Capability Exchange Attempts	Integer	Incremented whenever either side of the call initiates the capability exchange procedure (using either H.245 tunneling or a separate H.245 signaling path).
TERM_CAP_XCHG_SUCC_TOT	H.245 Terminal Capability Exchange Successes	Integer	Incremented whenever a capability exchange procedure is completed.
OPEN_CH_ATT_TOT	H.245 Open Logical Channel Attempts	Integer	Incremented whenever either side of the call initiates the open logical channel procedure (using either H.245 tunneling or a separate H.245 signaling path).
OPEN_CH_SUCC_TOT	H.245 Open Logical Channel Successes	Integer	Incremented whenever an open logical channel procedure is completed.
CLOSE_CH_ATT_TOT	H.245 Close Logical Channel Attempts	Integer	Incremented whenever either side of the call initiates the close logical channel procedure (using either H.245 tunneling or a separate H.245 signaling path).
CLOSE_CH_SUCC_TOT	H.245 Close Logical Channel Successes	Integer	Incremented whenever a close logical channel procedure is completed.
AVG_ROUND_TRIP_DELAY	H.245 Round Trip Delay Determination	Average (ms)	The average time in milliseconds (ms) for round trip delay measured as a result of successful round trip delay determination procedures.
EMPTY_CAP_SET_TOT	H245	Integer	Incremented each time an empty cap set request is received from the remote peer. Collection intervals are provisionable (default is 12 hours).
H323_T38_FAX_ATT_TOT	H245	Integer	Incremented for each T.38 Fax Call request from the remote peer. Collection intervals are provisionable (default is 12 hours)
H323_T38_FAX_SUCC_TOT	H245	Integer	Incremented for each T.38 Fax Call request from the remote peer that is successfully reconfigured for T.38 fax working. Collection intervals are provisionable (default is 12 hours).

Table 4-3 H.245 Counter Group (continued)

Counter Name	Measurement	Type	Comments
ASYMMETRIC_TOT	H245	Integer	Incremented for each asymmetric condition encountered. Collection intervals are provisionable (default is 12 hours).
DTMF_ RELAY_ TOT	H245	Integer	Incremented for each call where DTMF relay is used. Collection intervals are provisionable (default is 12 hours).

Resetting Measurements

The **clr-meas** MML command resets the measurement counters. This command resets an individual counter or all counters in a counter group. The following are valid counter groups:

- RAS
- Q.931
- H.245

For more information about the **clr-meas** command, see Appendix A, "MML User Interface and Command Reference."

Retrieving Counters

Use the **rtrv-ctr** MML command to retrieve measurement counters. This command displays the measurements for a counter group. Valid counter groups are RAS, Q.931, and H.245. For more information about the **rtrv-ctr** command, see Appendix A, "MML User Interface and Command Reference."

Overload

The system continuously checks call totals and CPU utilization. Each of these values is compared to predefined limits. Three call total limits are available. Each limit has a hysteresis value and an alarm associated with it. When the call total reaches the limit, an alarm is raised. When the call total falls below the limit minus the hysteresis value, the alarm is cleared after the appropriate recovery action is taken.

Cisco HSI supports the following three levels of overload:

- Overload level 1
- Overload level 2
- Overload level 3

The following factors can trigger any one of the overload levels:

- CPU usage (the OVLDSAMPLERATE parameter defines the frequency of CPU sampling and threshold checking)
- Maximum calls allowed

Disk usage can trigger a LOW_DISK_SPACE alarm. For more information about this alarm, see Chapter 5, "Troubleshooting Cisco HSI Alarms."

Overload Level 1

Use the following configuration parameters for overload level 1 (see Chapter 3, "Provisioning the Cisco HSI"):

- OVLDLEVEL1PERCENT
- OVLDLEVEL1FILTER
- OVLDLEVEL1THRESHLOWERCALLS
- OVLDLEVEL1THRESHUPPERCALLS
- OVLDLEVEL1THRESHLOWERCPU
- OVLDLEVEL1THRESHUPPERCPU

Overload Level 2

Use the following configuration parameters for overload level 2 (see Chapter 3, "Provisioning the Cisco HSI"):

- OVLDLEVEL2PERCENT
- OVLDLEVEL2FILTER
- OVLDLEVEL2THRESHLOWERCALLS
- OVLDLEVEL2THRESHUPPERCALLS
- OVLDLEVEL2THRESHLOWERCPU
- OVLDLEVEL2THRESHUPPERCPU

Overload Level 3

Use the following configuration parameters for overload level 3 (see Chapter 3, "Provisioning the Cisco HSI"):

- OVLDLEVEL3PERCENT
- OVLDLEVEL3FILTER
- OVLDLEVEL3THRESHLOWERCALLS
- OVLDLEVEL3THRESHUPPERCALLS
- OVLDLEVEL3THRESHLOWERCPU
- OVLDLEVEL3THRESHUPPERCPU

Setting Overload Data

The following MML commands set overload data:

set-overload:level1|level2|level3:cpu, lower=*number***, upper=***number*

set-overload:level1|level2|level3:calls, lower=*number***, upper=***number*

set-overload:level1|level2|level3:gap, filter=normal|all, percent=*number*

The upper parameter specifies the threshold for overload detection, and the lower parameter specifies the hysteresis point at which the overload condition is removed.

The lower value should be greater than the upper value of the next lower severity level.

For example:

set-overload:level1:cpu, lower=45, upper=50

set-overload:level1:gap, filter=normal, percent=50

set-overload:level2:cpu, lower=63, upper=70

set-overload:level2:gap, filter=normal, percent=75

set-overload:level3:cpu, lower=81, upper=90

set-overload:level3:gap, filter=normal, percent=95

These values mean that:

- At less than 50 percent CPU usage, no call is gapped.
- From 50 percent to 70 percent CPU usage, 50 percent of calls are gapped.
- From 70 percent to 90 percent CPU usage, 75 percent of calls are gapped.
- At more than 90 percent CPU usage, 95 percent of calls are gapped.
- Before the overload level returns from level 3 to level 2, the CPU usage must fall to less than 81 percent.

Note The HSI sends a Release message to the PGW when gapping calls. The cause value is derived from the property CCPackage,A_CC_GAPPEDCALLCAUSE, which is set to 60 (Congestion) in the default configuration. Cisco recommends configuring the Cisco PGW2200 dial plan to reroute the call when it receives this release cause.

Refer to the *Cisco Media Gateway Controller Software Release 9 Provisioning Guide* for further information.

Retrieving Overload Data

Use the **rtrv-overload** MML command to display the overload status and related overload data. For information about this command, see Appendix A, “MML User Interface and Command Reference.”

Logging

The logging level of one or more service packages is set using the **set-log** MML command. For more information about this command, see Appendix A, "MML User Interface and Command Reference."

Rotating Log Files

Log files are rotated at system startup or when either of the following conditions occurs:

- The size limit for the corresponding file is reached. The size of the corresponding log file is equal to or greater than the value that the LOGFILEROTATESIZE configuration parameter specifies. The default value for this parameter is 10 Mb (see Chapter 3, "Provisioning the Cisco HSI").
- The age limit for the corresponding file is reached. The corresponding log file is equal to or older than the interval that the LOGFILEROTATEINTERVAL parameter specifies. The default value for this parameter is 1440 minutes (24 hours). See Chapter 3, "Provisioning the Cisco HSI," for more information about this parameter.

Convention for Naming the Log File

Log rotation occurs when the system ceases to write to the current log file and commences to write to a new log file. The LOGFILENAMEPREFIX parameter defines the name of the active log file (see Chapter 3, "Provisioning the Cisco HSI"). The default is platform.log.

When log rotation is triggered, the existing file (for example, platform.log) is renamed with the format *platform_yyyymmddhhmmss.log* (see Table 4-4). For example, a platform error file rotated on September 30, 1999 at 12:36:24 is renamed platform_19990930123624.

Table 4-4 Log Filename Format

Format	Definition
LOGFILENAMEPREFIX	Provisioned filename (default is platform.log)
yyyy	Year
mm	Month
dd	Day
hh	Hour
mm	Minute
ss	Second

Note The time stamp is the coordinated universal time (CUT) from the machine at the time of rotation.

Log File Location

The LOGDIRECTORY parameter defines the directory for active log files and rotated log files (see Chapter 3, "Provisioning the Cisco HSI"). The default is $GWHOME/var/log/.

Log Messages

Log messages have the following format:

Date and timestamp, Package Name, <log level>, LogID:<text of the message>.

The following are examples of log messages:

```
Thu Dec  7 03:55:32:837 2000, Infrastructure, <DEBUG>, 205: GWModule Registration -
shutdownList() - NbOfItems 10 - Item 8
Thu Dec  7 03:55:32:837 2000, Infrastructure, <DEBUG>, 206 : GWModuleRegistration -
shutdownList() - NbOfItems 10 - Item 9
Thu Dec  7 03:55:32:838 2000, Infrastructure, <DEBUG>, 207 : GWReactor::thdId() returns 6.
Thu Dec  7 03:55:32:838 2000, Infrastructure, <DEBUG>, 208 : GWReactorModule::shutdown() -
Thread has joined.
```

Log Message Packages

The following service packages can log messages:

- Application
- CallControl
- Connection
- DataManager
- Eisup
- FaultManager
- Gapping
- H323
- Infrastructure
- Overload
- ProcessManager
- Provisioning
- Signal
- Snmp
- SnmpSubagent
- Statistics
- Trace
- UserInterface

Logging Levels

Logging levels determine how much debug information is stored in the platform.log file for each package. Levels are set through use of a hexadecimal number between 0x0000 and 0xFFFF. 0x0000 is the lowest level, and switches off logging for a particular package. 0xFFFF is the highest logging level.

Note We strongly recommend that you set all packages to log level 0x0000 in a live network. Set them to higher levels only when you debug on an offline network.

Setting Logging Levels

The **set-log** MML command dynamically alters the log level setting during the execution of the system. However, the **set-log** MML command does not affect the logging level of any current MML processes. For more information about the **set-log** command, see Appendix A, "MML User Interface and Command Reference."

Note The enabling of logging severely impacts HSI performance. We recommend the HSI be running at less than 2 calls per second when you enable logging. Logging will be automatically disabled when the HSI enters overload level 3. You can reenable logging when the HSI exits overload.

RADVision Logging

The Cisco HSI application provides the capability (through MML) to initiate RADVision logging. The contents of the resultant log file are not under the control of the Cisco HSI application.

Use the **radlog** MML command to start and stop RADVision logging. RADVision logging can be directed to a file or into the standard logging output. For information about this command, see Appendix A, "MML User Interface and Command Reference."

Gapping

The gapping level can be set from 0 to 100 percent. From 0 to 99 percent, the call type (normal or priority) is checked against the gapping level call status type. At 100 percent gapping, all calls are gapped regardless of call type.

Setting Gapping

To activate call gapping, complete the following steps:

Step 1 Determine the direction of the call to be gapped:

- Incoming (inc) for calls originating from the H.323 network
- Outgoing (otg) for calls originating from the PSTN Gateway (PGW 2200)
- Both (both) for calls originating from either side

Step 2 Determine what type of calls are to be gapped:

- Normal calls (nonpriority calls)
- All calls

Step 3 Determine the percentage of calls to be gapped. The percentage can range from 0 to 100 percent. If 100 percent is selected, all calls are gapped, regardless of the type of call.

Step 4 Enter the **set-gapping** MML command. For example, to gap 60 percent of all calls for both directions, enter:

set-gapping:both:calltype=all,percent=60

Retrieving Call Gapping Data

To retrieve the current levels of call gapping for all gapping clients, enter the **rtrv-gapping** command. The command displays text similar to the following:

```
Client Name      Direction        Level            Call Type        Active

Overload         Outgoing         10               Normal           No
Overload         Incoming         10               Normal           No
MML              Outgoing         20               All              Yes
MML              Incoming         30               All              Yes
```

The output shows the gapping levels set by the overload function and the MML command **set-gapping**. The highest gapping level is used as the level to gap calls, which is indicated as Yes in the column titled Active. In this example, the MML levels for outgoing and incoming calls are active.

Troubleshooting Cisco HSI Alarms

Introduction

This chapter contains information about Cisco H.323 Signaling Interface (HSI) alarms, troubleshooting procedures for these alarms, and information about detailed logging. This chapter contains the following sections:

Alarms Overview

An alarm can be in one of the following states:

- Raised, when a persistent fault occurs in the system
- Cleared, when the fault is fixed

Debounce

The alarms have a timeout (debounce) period. The debounce period is the time that elapses before an alarm condition is accepted. Use the ALARMDEBOUNCETIME parameter to set the debounce period (see Chapter 3, “Provisioning the Cisco HSI”). The default debounce period is 0.

Alarm Severity Levels

The Cisco HSI generates autonomous messages, or events, to notify you of problems or atypical network conditions. Depending on the event severity level, events are considered alarms or informational events. Table 5-1 lists the severity levels and the required responses.

Table 5-1 Alarm Severity Levels

Severity Level	Description
Critical	A serious problem exists in the network. Clear critical alarms immediately. A critical alarm should force an automatic restart of the application.
Major	A disruption of service has occurred. Clear this alarm immediately.
Minor	No disruption of service has occurred, but clear this alarm as soon as possible.
Informational	An abnormal condition has occurred. It is transient and does not require corrective action. (An invalid protocol call state transition is an example of an event that prompts such an alarm.) No corrective action is required by the management center to fix the problem.

Retrieving and Reporting Alarms

Events with a severity level of critical, major, or minor are classified as alarms and can be retrieved through the Man-Machine Language (MML) interface and a Simple Network Management Protocol (SNMP) manager.

An alarm must be reported when an alarm state changes (assuming the alarm does not have an unreported severity).

Informational Event Requirements

Informational events do not require state changes. An informational event is a warning that an abnormal condition that does not require corrective action has occurred. An invalid protocol call state transition is an example of an informational event. The informational event needs to be reported, but it is transient. No corrective action is required by the management center to fix the problem.

An informational event is reported once, upon occurrence, through the MML and SNMP interfaces. The MML interface must be in the **rtrv-alms:cont** mode for the event to be displayed. The event is not displayed in subsequent **rtrv-alms** commands.

SNMP Trap Types

Alarms have SNMP trap types associated with them. Table 6-2 identifies the trap types.

Table 5-2 SNMP Trap Types

Trap Type	Trap Description
0	No error
1	Communication alarm
2	Quality of service
3	Processing error
4	Equipment error
5	Environment error

Retrieving Alarm Messages

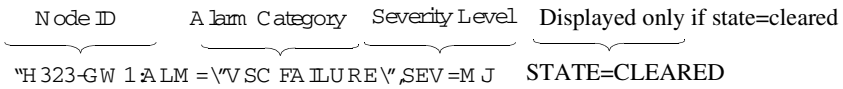
Alarms can be displayed in noncontinuous mode or in continuous mode.

Noncontinuous Mode

To display all current alarms, use the **rtrv-alms** MML command.

Figure 5-1 shows an example of an alarm message displayed with the **rtrv-alms** MML command (noncontinuous mode). For more information about the **rtrv-alms** MML command, see Appendix A, "MML User Interface and Command Reference."

Figure 5-1 Sample Alarm Message



The example in Figure 5-1 shows a Cisco Public Switched Telephone Network (PSTN) Gateway (PGW 2200) communication failure on the Cisco HSI that has the node ID H323-GW1. The resulting message is an alarm with a *major* severity level.

Continuous Mode

To display the names of active alarms and new alarm events, use the **rtrv-alms:cont** MML command.

Table 5-3 defines the message components that are displayed when the **rtrv-alms:cont** MML command is used. The following is sample output from this command. For more information about the **rtrv-alms:cont** MML command, see Appendix A, "MML User Interface and Command Reference."

```
GW Signaling Gateway    2000-12-05 14:19:22
M   RTRV
"H323-GW1: 2000-11-27 11:25:12.259, ** ALM=\"VSC FAILURE\",SEV=MJ"
"H323-GW1: 2000-11-27 11:25:13.259,    ALM=\"VSC FAILURE\",SEV=MJ"STATE=CLEARED
"H323-GW1: 2000-11-27 11:25:13.260, ** ALM=\"CONFIGURATION FAILURE\",SEV=MJ"
"H323-GW1: 2000-11-27 11:25:14.011, A^ ALM=\"ENDPOINT CHANNEL INTERFACE FAILURE\",SEV=IF"
"H323-GW1: 2000-11-27 11:25:14.012, A^ ALM=\"ENDPOINT CHANNEL INTERFACE FAILURE\",SEV=IF"

/* Listening for alarm events... (Ctrl-C to stop) */

"H323-GW1: 2000-11-27 11:25:13.259, ** ALM=\"VSC FAILURE\",SEV=MJ"

/* Ctrl-C pressed */
```

Table 5-3 Elements of Continuous Mode Messages

Element	Description
systemId	The name of your device and its identifier.
YYYY-MM-DD	The year, month, and day that the alarm or information event occurred.
hh-mm-ss-ms	The hour, minute, second, and millisecond that the alarm or information event occurred.
severity	The severity level of the alarm or information event. Severity is represented by a two-character indicator with the following meanings: • *C—Critical alarm. A critical alarm indicates that a serious problem exists in the network. It causes a restart or reboot of the Cisco HSI. Clear critical alarms immediately. • **—Major alarm. A major alarm indicates the existence of a problem that disrupts service. Clear major alarms immediately. Major alarms differ from critical alarms in that they do not initiate automatic recovery processes. • *^—Minor alarm. A minor alarm indicates the presence of a problem that does not disrupt service. Note and clear minor alarms as soon as possible. • A^—Informational event. An informational event indicates the presence of an atypical network condition, such as a timer expiration, a value that has exceeded preset thresholds, or unexpected response from an end point to a signaling messages sent by the Cisco HSI. • — (Empty spaces in two leftmost columns). The alarm or event has been cleared. “STATE=CLEARED” is displayed.
almCat	Alarm category. A text string that indicates whether the message is an alarm or an informational event and the MML alarm or event message. See Table 5-4 for a list of alarm categories. **Note** Despite its name, the alarm category field is used for both alarms and informational events.
Acknowledgement	Indicates whether the alarm has been acknowledged.

Acknowledging and Clearing Alarms

To acknowledge that an alarm is recognized but not cleared, use the **ack-alm** MML command. See Appendix A, “MML User Interface and Command Reference,” for more information.

To clear an alarm, use the **clr-alm** MML command. See Appendix A, “MML User Interface and Command Reference,” for more information.

Alarms List

Table 5-4 lists alarms and information events. Troubleshooting information for each of the alarms and information events can be found in the "Troubleshooting" section on page 5-6.

Table 5-4 Alarms and Informational Events

Alarm Event and Reference	Severity Level
H323_STACK_FAILURE, page 5-6	Critical
CONFIGURATION_FAILURE, page 5-6	Major
EISUP_PATH_FAILURE, page 5-7	Major
GATEKEEPER_INTERFACE_FAILURE, page 5-8	—
GENERAL_PROCESS_FAILURE, page 5-8	Major
IP_LINK_FAILURE, page 5-8	Major
LOW_DISK_SPACE, page 5-9	Major
OVERLOAD_LEVEL3, page 5-9	Major
VSC_FAILURE, page 5-10	Major
OVERLOAD_LEVEL2, page 5-11	Minor
CONFIG_CHANGE, page 5-11	Information
ENDPOINT_CALL_CONTROL_INTERFACE_FAILURE, page 5-12	Information
ENDPOINT_CHANNEL_INTERFACE_FAILURE, page 5-12	Information
GAPPED_CALL_NORMAL, page 5-13	Information
GAPPED_CALL_PRIORITY, page 5-13	Information
OVERLOAD_LEVEL1, page 5-14	Information
PROVISIONING_INACTIVITY_TIMEOUT, page 5-14	Information
PROVISIONING_SESSION_TIMEOUT, page 5-15	Information
STOP_CALL_PROCESSING, page 5-15	Information

Troubleshooting

This section provides troubleshooting procedures for the alarms listed in Table 5-4.

H323_STACK_FAILURE

Description

Irrecoverable failure in the RADVision stack. This alarm is reported to the management interface and can be obtained with SNMP.

Severity Level and Trap Type

The severity level is critical. The trap type is 4.

Cause

The H.323 RADVision stack has failed to correctly initialize on an application startup. An automatic application restart is initiated, and the application reverts to the base configuration data .

Troubleshooting

To clear the H.323 stack failure alarm, complete the following steps:

Step 1 Allow the application to restart and revert back to the base configuration data that is known to be reliable.

Step 2 Review the H323_SYS parameters in a provisioning session, ensuring that the values are correct and within the memory limits of the machine.

Step 3 Use the **prov-cpy** MML command to recommit the new H323_SYS parameters.

Step 4 Use the **restart-softw** MML command to initiate a software restart.

Step 5 Use the **rtrv-alms** MML command to check the alarm list to see if the H.323 stack correctly initializes.

CONFIGURATION_FAILURE

Description

The configuration has failed. This alarm is reported to the management interface and can be obtained with SNMP.

Severity Level and Trap Type

The severity level is major. The trap type is 4.

Cause

A major error has occurred in the configuration of the software packages. This is a potentially nonrecoverable situation that requires an application restart.

Troubleshooting

To clear the CONFIGURATION_FAILURE alarm, complete the following steps:

Step 1 Use the **restart-softw:init** command to restart the application and revert to the base configuration.

Step 2 Review the modified parameters and ensure that the values are correct.

Step 3 Use the **prov-cpy** MML command to recommit the new parameters.

Step 4 Use the **restart-softw** MML command to initiate a software restart.

Step 5 Use the **rtrv-alms** MML command to check the alarm list to see if the problem has been resolved.

EISUP_PATH_FAILURE

Description

A failure of the RUDP layer has occurred. This alarm is reported to the management interface and can be obtained with SNMP.

Severity Level and Trap Type

The severity level is major. The trap type is 4.

Cause

Both IP links A and B to a single Cisco PGW 2200 have gone down.

Troubleshooting

To clear the EISUP_Path_Failure alarm, complete the following steps:

Step 1 Use the **rtrv-dest** command to assess which Cisco PGW 2200 (standby or active) has been lost.

Step 2 Check the network connections, cables, and routers for that system.

Step 3 Use the **clr-alm** MML command to attempt to clear the alarm.

GATEKEEPER_INTERFACE_FAILURE

This alarm has not been implemented.

GENERAL_PROCESS_FAILURE

Description

A general process failure has occurred. This alarm is reported to the management interface and can be obtained with SNMP.

Severity Level and Trap Type

The severity level is major. The trap type is 4.

Cause

The Cisco HSI (GWmain program) quit unexpectedly (that is, there were no requests to stop or restart the application). The process manager (PMmain) raises the GENERAL_PROCESS_FAILURE alarm so that a trap is sent to the Cisco Media Gateway Controller Node Manager.

The process manager clears the GENERAL_PROCESS_FAILURE alarm when it restarts the Cisco HSI (GWmain).

Troubleshooting

To trace the problem, look at either the core file or the log files.

IP_LINK_FAILURE

Description

A failure of the IP link has occurred. This alarm is reported to the management interface and can be obtained with SNMP.

Severity Level and Trap Type

The severity level is major. The trap type is 4.

Cause

One of the two links to a single Cisco PGW 2200 has failed.

Troubleshooting

To clear the IP link failure alarm, complete the following steps:

Step 1 Use the **rtrv-dest** command to assess which PGW 2200 (standby or active) has been lost.

Step 2 Check the network connections, cables, and routers for that system.

Step 3 Use the **clr-alm** MML command to attempt to clear the alarm.

LOW_DISK_SPACE

Description

The disk space is low. This alarm is reported to the management interface and can be obtained with SNMP. The alarm automatically clears when the disk usage decreases below the alarm limit.

Severity Level and Trap Type

The severity level is major. The trap type is 4.

Cause

The percentage of disk usage is greater than the alarm limit.

Troubleshooting

To obtain more disk space, remove old versions of installed software that are no longer required, or archive log files from the $GWHOME/var/log directory, for example.

OVERLOAD_LEVEL3

Description

An overload level 3 condition exists. This alarm is reported to the management interface and can be obtained with SNMP. This alarm automatically clears when the CPU occupancy or the number of active calls drops below the lower limits set in the overload configuration for level 3.

Severity Level and Trap Type

The severity level is major. The trap type is 4.

Cause

The OVERLOAD_LEVEL3 alarm is triggered when the CPU occupancy or the number of active calls rises above the upper limits set in the overload configuration for level 3. Gapping is then initiated.

Troubleshooting

To clear the OVERLOAD_LEVEL3 alarm, complete the following steps:

Step 1 Wait for the number of calls to drop.

Step 2 If CPU occupancy remains high, request assistance from the system administrator.

VSC_FAILURE

Description

This alarm is derived by the Cisco HSI application from RUDP/SM events. This alarm is reported to the management interface and can be obtained with SNMP.

Severity Level and Trap Type

The severity level is major. The trap type is 5.

Cause

Links to both (active and standby) Cisco PGW 2200s have gone down.

Troubleshooting

To clear the VSC_FAILURE alarm, complete the following steps:

Step 1 Use the **rtrv-dest** command to confirm that links to the Cisco PGW 2200s have gone down.

Step 2 Check the network connections, cables, and routers.

Step 3 Refer to the *Cisco Media Gateway Controller Software Release 9 Operations, Maintenance, and Troubleshooting Guide* for detailed information about this alarm.

Step 4 Use the **clr-alm** command to attempt to clear the alarm.

OVERLOAD_LEVEL2

Description

An overload level 2 condition exists. This alarm is reported to the management interface and can be obtained with SNMP. This alarm automatically clears when the CPU occupancy or the number of active calls drops below the lower limits set in the overload configuration for level 2.

Severity Level and Trap Type

The severity level is minor. The trap type is 4.

Cause

The OVERLOAD_LEVEL2 alarm is triggered when the CPU occupancy or the number of active calls rises above the upper limits set in the overload configuration for level 2. Gapping is then initiated.

Troubleshooting

To clear the OVERLOAD_LEVEL2 alarm, complete the following steps:

Step 1 Wait for the number of calls to drop.

Step 2 If CPU occupancy remains high, request assistance from the system administrator.

CONFIG_CHANGE

Description

The running configuration has been modified.

Severity Level and Trap Type

The severity level is information. The trap type is 0.

Cause

A new configuration has been activated within a provisioning session.

Troubleshooting

This is an informational event.

ENDPOINT_CALL_CONTROL_INTERFACE_FAILURE

Description

An individual call failure has occurred. This informational event is reported to the management interface and can be obtained with SNMP.

Severity Level and Trap Type

The severity level is information. The trap type is 3.

Cause

The RADVision stack reports this alarm.

Troubleshooting

This is an informational event.

ENDPOINT_CHANNEL_INTERFACE_FAILURE

Description

An individual call failure has occurred. This informational event is reported to the management interface and can be obtained with SNMP.

Severity Level and Trap Type

The severity level is information. The trap type is 3.

Cause

The RADVision stack reports this alarm.

Troubleshooting

This is an informational event.

GAPPED_CALL_NORMAL

Description

A normal call has been rejected due to call gapping. This informational event is reported to the management interface and can be obtained with SNMP.

Severity Level and Trap Type

The severity level is information. The trap type is 2.

Cause

The GAPPED_CALL_NORMAL alarm is triggered when gapping levels cause a normal call to be rejected.

Troubleshooting

To clear the GAPPED_CALL_NORMAL informational event, complete the following steps:

Step 1 Use the **rtrv-gapping** MML command to retrieve gapping information.

Step 2 If the MML-specific gap levels are active, use the **set-gapping** MML command to modify them.

Step 3 If the overload-specific gap levels are active, either modify the provisioned overload gapping percent levels or reduce the cause of the overload (see OVERLOAD_LEVEL1, page 5-14, OVERLOAD_LEVEL2, page 5-11, and OVERLOAD_LEVEL3, page 5-9).

GAPPED_CALL_PRIORITY

Description

A priority or emergency call has been rejected due to call gapping. This informational event is reported to the management interface and can be obtained with SNMP.

Severity Level and Trap Type

The severity level is information. The trap type is 2.

Cause

The GAPPED_CALL_NORMAL alarm is triggered when gapping levels cause a priority or emergency call to be rejected.

Troubleshooting

To clear the GAPPED_CALL_PRIORITY informational event, complete the following steps:

Step 1 Change the MML gapping levels to less than 100 percent and change the call type to normal.

Step 2 Change the provisioned overload call filter type to normal.

OVERLOAD_LEVEL1

Description

An overload level 1 condition exists. This informational event is reported to the management interface and can be obtained with SNMP.

Severity Level and Trap Type

The severity level is information. The trap type is 4.

Cause

The OVERLOAD_LEVEL1 alarm is triggered when the CPU occupancy or the number of active calls rises above the upper limits set in the overload configuration for level 1. Gapping is then initiated.

Troubleshooting

To clear the OVERLOAD_LEVEL1 informational event, complete the following steps:

Step 1 Wait for the number of calls to drop.

Step 2 If CPU occupancy remains high, request assistance from the system administrator.

PROVISIONING_INACTIVITY_TIMEOUT

Description

A provisioning session has been inactive for 20 minutes. The text of the output is:

```
"H323-GW1:2001-01-30 11:12:57.421,A^ ALM=\"PROVISIONING INACTIVITY TIMEOUT\",SEV=IF"
```

Severity Level and Trap Type

The severity level is information. The trap type is 3.

Cause

The provisioning session has been inactive for 20 minutes. The provisioning session will be closed if there is no activity within the next 5 minutes.

Troubleshooting

Ensure that activity in the provisioning session occurs at least every 20 minutes.

PROVISIONING_SESSION_TIMEOUT

Description

The current session has been terminated. The text of the output is:

```
"H323-GW1:2001-01-30 11:17:57.422,A^ ALM=\"PROVISIONING SESSION
TIMEOUT\",SEV=IF"
```

Severity Level and Trap Type

The severity level is information. The trap type is 3.

Cause

The provisioning session has been inactive for longer than the time allowed.

Troubleshooting

Ensure that activity within the provisioning session occurs at least every 20 minutes.

STOP_CALL_PROCESSING

Description

A stop call processing request has been entered through the MML.

Severity Level and Trap Type

The severity level is information. The trap type is 4.

Cause

A user has entered the **stp-callproc** command through the MML.

Troubleshooting

This is an informational event.

Detailed Logging

Logging occurs on 16 different levels for each package, and the logging mask (which is a 16-bit number from 0x0000 to 0xFFFF) allows each specific log level to be turned on and off. The most-significant-bit positions correspond to higher (that is, more processor intensive) levels of debugging.

We recommend that you set the logging level of all packages to 0x0000 in a live network. For debugging a single call in an off-line network, the recommended level of debug is:

- Set Eisup, CallControl, and H323 package log levels to 0xFFFF.
- Set all other package log levels to 0x0000.
- Turn radlog on by entering the MML command **radlog::start**.

Once the test call has been made, remember to set all the logging levels back to 0x0000 and to turn radlog off by entering the MML command **radlog::stop**.

MML User Interface and Command Reference

Introduction

This appendix provides information about Man-Machine Language (MML) command syntax and conventions, batch files, and procedures for starting and stopping MML sessions in the Cisco H.323 Signaling Interface (HSI) application. The appendix contains the following sections:

Starting an MML Command Session in the Cisco HSI

To start an MML command session within the HSI environment, complete the following steps:

Step 1 Log in to Cisco HSI as mgcusr.

Step 2 Become superuser by typing **su** <*root password*>.

Step 3 To start the software, enter the following command:

/etc/init.d/CiscoGW start

Step 4 Exit out of superuser.

Step 5 Type **mml** to start the MML command-line interpreter.

Tip Press the **Tab** key twice to see a list of MML commands.

MML Commands

To execute MML commands, log in to Cisco HSI and perform one of the following tasks:

- Start the MML session (see the "Starting an MML Session" section on page A-3) and enter a command.
- Type a batch file command to start an MML session (see the "Starting a Batch File" section on page A-4).

MML Command Syntax

MML commands use the following syntax:

command_name:target:[Parameter_List][;comments]

Parameter_List consists of a parameter name, an equal sign, and a value for the parameter.

The keywords and the value strings need not be enclosed in quotation marks. Anything you enter after a semicolon (;) is treated as a comment. Use only one MML command on each line.

The "MML Command Reference" section on page A-7 contains detailed information about the individual MML commands.

Tip

- Use the **Up Arrow** key to scroll through all previous MML commands in turn.
- Use the **Down Arrow** key to move forward in the command buffer.
- Use the **Left** and **Right Arrow** keys to move along the command line.
- Use the **Backspace**, **Delete**, and alphanumeric keys to edit an MML command.

MML Command Conventions

The MML commands use the conventions shown in Table A-1.

Table A-1 MML Command Conventions

Convention	Meaning	Comments and Examples
Square brackets ([ ])	Optional elements	**command** [abc] abc is optional (not required), but you can choose it.
Vertical bars (|)	Separated alternative elements	**command** [abc | def] You can choose either abc or def, or neither, but not both.
Braces ({ | })	Required choice of alternative elements	**command** {abc | def} You must use either abc or def, but not both.
Angle brackets (< >)	Symbol specifier	—

The MML commands can be interpreted and monitored through a network Transaction Language 1 (TL1) interface. The TL1 symbols shown in Table A-2 are used in MML

Table A-2 TL1 Symbols Used in MML

Symbol	Description
:	A parameter separator.
::	An empty parameter block.
&	Arguments are grouped together so that one parameter can convey several arguments.
;	End of command (optional). Anything on the same line after this symbol is treated as a comment.

Case Sensitivity

Command names and parameter names are not case sensitive You can enter commands and parameters in either upper- or lowercase. Filenames *are* case sensitive when they are used as arguments in MML commands.

Starting an MML Session

To start an MML session, complete the following steps:

Step 1 Log in to Cisco HSI.

Step 2 Type one of the following commands at the prompt:

- **mml**
- **mml -b batchfile** (see the "Starting a Batch File" section on page A-4)

The following example shows the start of an MML session:

```
user@host> mml

Welcome to the Cisco H.323 Signaling Interface.

gw mml>
```

Batch Files

The Cisco HSI application supports the use of batch files. You can create an ASCII file of MML provisioning commands for use as a batch file. You can also use a script file. When the commands are read, the Cisco HSI executes them sequentially.

The following is an example of an MML provisioning batch file:

```
prov-sta::srcver="new",dstver="first"
prov-add:name="sys_config",nodeid="H323-GW1"
prov-add:name="h323_sys",messages=30000,channels=5000
prov-add:name="ras",responsetimeout=10,allowcallswhennonreg=1
prov-add:name="ras",terminaltype=”gateway”,timetolive=900
prov-add:name="q931",reponsetimeout=20,connecttimeout=20,maxcalls=5000
prov-cpy
```

The **prov-sta** command establishes a provisioning session. The **prov-cpy** command copies configuration settings from the current provisioning session to the Cisco HSI and activates the configuration. If the command is successful, it also terminates the current provisioning session. If you are not ready to commit a session, use the **prov-stp** command to save and stop the provisioning session.

The application provides a log function (**diaglog** command) that records the MML commands and responses in a log file.

In the MML batch file, you can place a **diaglog** command at the beginning to start logging and a **diaglog** command at the end to stop logging. For more information about the **diaglog** command, see the “MML Command Reference” section on page A-7

For commands executed in both the process manager and the application, the application logs the user ID, the login date and time, and the name of each command that is executed in batch mode to the mml_batch_log file.

Note Batch files can be defined for complete systems or to modify parts of an existing system.

Creating a Batch File

To create a batch file, use an ASCII text editor program to create a new file. Each command should be on a separate line.

Starting a Batch File

To start executing a batch file, type **mml -b** *batchfilename* at the UNIX prompt.

After you enter the batch file command, the application displays the result of each MML command as it is executed. Each command and its results are saved in the mml.log file. When the batch file is completed, the MML session is ready to accept user commands.

The following example shows the start of a batch file named nolog.bat with these contents:

```
prov-sta:srcver=active,dstver=nolog
prov-ed:name=logging,eisup=0x0000
prov-cpy
```

and this output:

```
gp-capetown-16-> mml -b nolog.bat
Starting in batch mode.
Connecting to port 10129 on host gp-capetown

Welcome to the Cisco H.323 Signaling Interface.
gw mml> gw mml> prov-sta:srcver=active,dstver=nolog
H323 Signaling Interface Tue Jan 22 05:57:12 2002
M SUCC
Successfully started provisioning session "nolog" from "active".Note: This provisioning
session has not been verified.
gw mml> prov-ed:name=logging,eisup=0x0000
H323 Signaling Interface Tue Jan 22 05:57:12 2002
M SUCC
Successfully edited provisioning element(s):
MML Name : logging.
Parameter: EISUP.
Value : 0x0000.
gw mml> prov-cpy
H323 Signalling Gateway Tue Jan 22 05:57:13 2002
M SUCC
Successfully activated provisioning session nolog.
gw mml>
```

MML Responses

The following sections describe the two types of response messages that are displayed by the MML user interface:

- Status messages
- Error messages

Status Messages

Table A-3 lists the MML status messages and their descriptions.

Table A-3 MML Status Messages and Descriptions

Status	Message	Description
RTRV	Retrieve	Retrieve and display the contents of the specified file
SUCC	Successful	Successful completion

Error Messages

If an MML command does not perform, an error message is displayed. Table A-4 lists the MML error messages and their descriptions.

Table A-4 MML Error Messages and Descriptions

Error Message	Definition	Description
DENY	Command denied	The system recognizes the command but does not perform the requested function.
ICNV	Input command not valid	The system does not recognize the MML command.
IDNV	Input data not valid	An unknown parameter was entered.
IISP	Input syntax error	Incorrect syntax was entered.
IITA	Invalid target	The system cannot perform the requested operation on the specified component, or the component does not exist.
IPRM	Input parameter missing	An expected parameter was not entered.
SABT	Status abort	The requested operation did not complete within the allotted time.
SNVS	Component not in valid state	The requested operation failed because the component is either not configured to accept the operation or the component is already in the desired state.
SNSP	State not supported	The operation is not supported by the component.
SROF	Status requested operation failed	The requested operation failed.

MML Help

MML has an online help feature. The MML **help** command displays a list of valid system commands and an explanation of each command's use. To display the online help, start an MML session and type **help** at the command line prompt. See the "help" section on page A-11.

Quitting an MML Session

To quit an MML session, type **quit** at the prompt.

MML Command Reference

This section describes the following MML commands:

- **ack-alm**
- **clr-alm**
- **clr-meas**
- **diaglog**
- **h**
- **help**
- **prov-add**
- **prov-cpy**
- **prov-dlt**
- **prov-ed**
- **prov-exp**
- **prov-rtrv**
- **prov-sta**
- **prov-stp**
- **quit**
- **radlog**
- **restart-softw**
- **rtrv-alms**
- **rtrv-calls**
- **rtrv-ctr**
- **rtrv-dest**
- **rtrv-gapping**
- **rtrv-log**
- **rtrv-mml**
- **rtrv-ne**
- **rtrv-ne-health**
- **rtrv-overload**
- **rtrv-softw**
- **set-dest-state**
- **set-gapping**
- **set-log**
- **set-overload**
- **sta-callproc**
- **sta-softw**
- **sta-trc**

- **stp-call**
- **stp-callproc**
- **stp-softw**
- **stp-trc**

ack-alm

The **ack-alm** command acknowledges that an alarm event is recognized but does not clear the alarm.

ack-alm:event=*alarm event*

Syntax Description

alarm event	The alarm category or the text that appears in the body of the alarm. Alarm event names are defined in Chapter 5, "Troubleshooting Cisco HSI Alarms."

Usage Guidelines

The format of the alarm category name must be the same as the format of the alarm category name that the **rtrv-alms** command displays. It is case sensitive.

Examples

This example recognizes the **VSC_FAILURE** alarm event is recognized, but the alarm is not cleared:

```
gw mml> ack-alm:event=VSC_FAILURE

GW Signaling Interface    2000-12-05 14:19:22
M   SUCC

mml>
```

Related Commands

Command	Description
clr-alm	Clears an alarm event
rtrv-alms	Displays all active alarms

clr-alm

The **clr-alm** command clears an alarm event.

clr-alm:event=*alarm event*

Syntax Description

alarm event	The alarm event name or the text that appears in the body of the alarm. Alarm names are defined in Chapter 5, "Troubleshooting Cisco HSI Alarms."

Usage Guidelines

The alarm category must match the format shown in the alarm when the **rtrv-alms** command displays it. It is case sensitive.

Examples

This example clears the alarm event **VSC_FAILURE**.

```
gw mml> clr-alm:event=VSC_FAILURE

GW Signaling Interface    2000-12-05 14:19:22
M   SUCC

mml>
```

Related Commands

Command	Description
ack-alm	Acknowledges that an alarm event is recognized but does not clear the alarm.
rtrv-alms	Displays all active alarms

clr-meas

The **clr-meas** command resets a measurement counter.

clr-meas:*counter group***:name=***measurement name*

clr-meas:*counter group*

Syntax Description

counter group	Valid counter groups are: • RAS • Q931 • H245
measurement name	For a list of measurement names, see Table 4-1, Table 4-2, and Table 4-3.

Examples

In this example, a measurement counter, GK_DISC_ATT_TOT (Gatekeeper Discovery Attempts), is reset in the counter group RAS:

```
gw mml> clr-meas:RAS

GW Signaling Interface    2000-12-05 14:19:22
M   SUCC
mml>

mml> clr-meas:RAS:name=GK_DISC_ATT_TOT

GW Signaling Interface    2000-12-05 14:19:22
M   SUCC
```

Related Commands

Command	Description
rtrv-ctr	Displays the measurements for a counter group.

diaglog

The **diaglog** command starts and stops event logging into a diagnostics log.

diaglog:*file name***:start | stop**

Syntax Description

file name	The user-defined name of the log file. The actual file name has a .log suffix applied. The file is located in the logging directory defined in the configuration data (see Chapter 3, "Provisioning the Cisco HSI").

Examples

In this example, event logging is started in a diagnostics log named **test5**.

```
gw mml> diaglog:test5:start
test5_davek15823_20010130053323.log
```

In the preceding example, davek is the user who runs the command, and 15823 is the process ID of the MML process from which the command is run.

Related Commands

Command	Description
radlog	Starts and stops RADVision logging into a specified log file.
rtrv-log	Displays the logging level of a package or all packages.
set-log	Sets the logging level of a package or all packages.

h

The **h** command redisplays a command or a series of commands. Items displayed depend on a specified number or range. If no number is specified, only the last command is displayed.

h[::**start**=*number*[,**end**=*number*]]

Syntax Description

start	Entered as a number; specifies the first command to redisplay.
end	Entered as a number; specifies the end of the range of commands to redisplay.

Examples

The MML command in the following example displays the last successful command entered:

```
mml> h
VSC H-323 Signaling Interface - H323-GW1 2000-06-20 10:04:28
M RTRV
   "rtrv-log:all"
   /* command 1 */
```

The MML command in the following example displays the third from the last successful command entered:

```
mml> h::3
VSC H-323 Signaling Interface - H323-GW1 2000-06-20 10:04:28
M RTRV
   "rtrv-ne"
   /* command 3 */
```

The MML command in the following example displays the last and second to last commands entered.

```
mml> h::start=1,end=2
VSC H-323 Signaling Interface - H323-GW1 2000-06-20 10:04:28
M RTRV
   "rtrv-log:all"
   /* command 1 */
   "rtrv-ne"
   /* command 2 */
```

help

The **help** command displays a list of valid system commands and an explanation of their use. If you do not enter a command name as a parameter, the **help** command provides a list of MML commands, descriptions, and values. If you enter a command name as a parameter, a description of that command displays.

help[**:***command name*]

Syntax Description

command name	The name of the MML command.

Examples

The command shown in the following example displays help for a specific command:

```
mml> help:rtrv-ctr
H323 Signalling Gateway   Tue Feb 12 19:09:58 2002
M SUCC

                  RTRV-CTR -- Display the measurements for a counter group
                  ----------------------------------------------------------

   Purpose:     This MML command displays a measurement counter for a counter group

   Format:      rtrv-ctr:<counter group>

   Description: * counter group -- The name of an MML counter group (RAS, Q931 or H245)

   Example:     The MML command shown in the following example displays measurement
                counters for the counter group RAS.
                mml> rtrv-ctr:ras;
                   GW Signalling GateWay 2000-12--5 14:19:32
                M RTRV
                "H323-GW1:GROUP=RAS,NAME=\"GK_DISC_ATT_TOT\",VAL=1000"
                "H323-GW1:GROUP=RAS,NAME=\"GK_REG_ATT_TOT\",VAL=1000"
                "H323-GW1:GROUP=RAS,NAME=\"GK_REG_SUCC_TOT\",VAL=1000"

                mml>
```

If you enter the **help** command without a parameter, the help file displays information about all available commands. The following example shows a portion of the help file that displays if you do not enter a parameter:

```
mml> help

VSC H323 signaling interface – H323-GW1 2000-06-20 10:04:28
M RTRV
Available commands (in alphabetical order):
ack-alm:”<alm cat>” Acknowledges an alarm category on a component
clr-alm:”<alm cat>” Clears an alarm category on a component
clr-meas:”<meas cat>” Resets a measurement category on a component
diaglog:<file name>:START|STOP Starts/stops diagnostics log
h[::<number>[,<number>]] Displays a history of commands for a specified backward number or
range; the last command by default
help[:<command name>] Displays the list of MML commands or the help information on a
specified command
prov-add:name=<MML name>,<param name>=<param value>,... Adds the component
prov-cpy Commits provisioning data
prov-dlt:name=<MML name> Deletes the component
```

prov-add

The **prov-add** command adds a component to the Cisco HSI.

prov-add:name=*MML name*,*param name*=*param value*,...

Syntax Description

MML name	MML name for the element you are adding. Valid MML names are: • sys_config_static • sys_config_dynamic • h323_sys • ras • h245 • q931
param name	The name of a valid configuration parameter for the specified name.
param value	The value you want to assign to the parameter.

Usage Guidelines

To define more than one parameter, enter additional *param name=param value* descriptions on the command line. See Chapter 3, "Provisioning the Cisco HSI," for the list of MML names, parameter names, and their associated values.

Examples

The command shown in the following example adds a provisioning element with the MML name ras, parameter name maxFail, and value 3:

```
gw mml> prov-add:name=ras,maxfail=3
H323 Signaling Interface Sun Jan 7 15:15:02 2001
M SUCC
Successfully added provisioning element(s):
MML Name : ras.
Parameter: maxFail.
Value : 3.
```

Related Commands

Command	Description
prov-cpy	Activates the configuration settings in the current provisioning session.
prov-dlt	Deletes a provisioned component.
prov-ed	Modifies a provisioned component.
prov-exp	Exports the current configuration of the Cisco HSI in MML command form to a file or files.
prov-rtrv	Retrieves information about an existing provisioning session.
prov-sta	Establishes a provisioning session.
prov-stp	Terminates either a specified provisioning session or the current provisioning session.

prov-cpy

The **prov-cpy** command activates the current provisioning session. If any client-level parsing fails during the **prov-cpy** command, the system might prompt for confirmation to force the HSI to activate the configuration (**prov-cpy[:confirm]**). However, we recommend that you never use **prov-cpy:confirm** unless you are asked to do so by Cisco.

If client-level parsing fails, there is a severe error within the user configuration.

If the **prov-cpy** command fails, use the failure description and the configuration changes to determine the error. Correct the configuration and reissue the **prov-cpy** command.

Please contact the Cisco TAC if you require assistance with the configuration.

Syntax Description

confirm	If any client level parsing fails during the data session, a **confirm** is needed for the data to be activated.

Examples

The command shown in the following example copies the configuration changes from the current session to the Cisco HSI:

```
gw mml> prov-cpy
H323 Signaling Interface Sun Jan 7 13:53:42 2001
M SUCC
Successfully activated the provisioning session.
```

Usage Guidelines

See Chapter 3, "Provisioning the Cisco HSI," for a list of MML names, parameter names, and their associated values.

Related Commands

Command	Description
prov-add	Adds a component.
prov-dlt	Deletes a provisioned component.
prov-ed	Modifies a provisioned component.
prov-exp	Exports the current configuration of the Cisco HSI in MML command form to a file or files.
prov-rtrv	Retrieves information about an existing provisioning session.
prov-sta	Establishes a provisioning session.
prov-stp	Terminates either a specified provisioning session or the current provisioning session.

prov-dlt

The **prov-dlt** command deletes a provisioned component. It allows you to delete a parameter rather than deleting the MML group.

prov-dlt:name=*MML name*

prov-dlt:name=*MML name***,param=***param name*

prov-dlt:name=*MML name param name*

Syntax Description

MML name	MML name for the element you are deleting. Valid MML names are: • sys_config_static • sys_config_dynamic • h323_sys • ras • h245 • q931
param name	The name of a valid configuration parameter for the specified name.

Usage Guidelines

See Chapter 3, "Provisioning the Cisco HSI," for a list of MML names, parameter names, and their associated values.

Examples

The MML command in the following example deletes the ras element:

```
gw mml> prov-dlt:name=ras
H323 Signaling Interface Sun Jan 7 14:13:05 2001
M SUCC
Successfully deleted provisioning data for ras
```

The MML command in the following examples delete the maxCalls parameter of the ras element:

```
gw mml> prov-dlt:name=ras,param=maxCalls
gw mml> prov-dlt:name=ras,maxCalls
H323 Signaling Interface Sun Jan 7 14:46:01 2001
M SUCC
Successfully deleted provisioning data for ras:maxCalls
```

Related Commands

Command	Description
prov-add	Adds a component.
prov-cpy	Activates the configuration settings in the current provisioning session.
prov-ed	Modifies a provisioned component.
prov-exp	Exports the current configuration of the Cisco HSI in MML command form to a file or files.

Command	Description
prov-rtrv	Retrieves information about an existing provisioning session.
prov-sta	Establishes a provisioning session.
prov-stp	Terminates either a specified provisioning session or the current provisioning session.

prov-ed

The **prov-ed** command modifies a provisioned component.

prov-ed:name=*MML name*,*param name*=*param value*,...

Note Enter only those parameters that must be modified.

Syntax Description

MML name	MML name for the element you are modifying. Valid MML names are: • sys_config_static • sys_config_dynamic • h323_sys • ras • h245 • q931
param name	The name of a valid configuration parameter for the specified name.
param value	The value you want to assign to the parameter.

Usage Guidelines

To change more than one parameter, enter additional *param name=value* descriptions on the command line. See Chapter 3, "Provisioning the Cisco HSI," for a list of MML names, parameter names, and their associated values.

Examples

Use the MML command shown in the following example to edit a provisioning element with the MML name ras, parameter name maxFail, and value 7:

```
gw mml> prov-ed:name=ras,maxfail=7
H323 Signaling Interface Sun Jan 7 15:22:02 2001
M SUCC
Successfully edited provisioning element(s):
MML Name : ras.
Parameter: maxFail.
Value : 7.
```

Related Commands

Command	Description
prov-add	Adds a component.
prov-cpy	Activates the configuration settings in the current provisioning session.
prov-dlt	Deletes a provisioned component.
prov-exp	Exports the current configuration of the Cisco HSI in MML command form to a file or files.
prov-rtrv	Retrieves information about an existing provisioning session.
prov-sta	Establishes a provisioning session.
prov-stp	Terminates either a specified provisioning session or the current provisioning session.

prov-exp

The **prov-exp** command exports the current provisioned configuration of the Cisco HSI in MML command form to a file. With this configuration file, you can prime a system with a cloned configuration from an existing system. It also enables you to restore a baseline configuration to a system. You can use the MML batch feature to import the exported data.

Start a dummy provisioning session with the **prov-sta** command before you use the **prov-exp** command.

prov-sta:srcver=active, dstver=dummy1

prov-exp:dirname=*export directory name*

prov-stp

Syntax Description

export directory name	Name of the directory to which the data is exported. This directory is a subdirectory within the /opt/GoldWing/export directory established at installation.

Examples

The MML command shown in the following example saves the active file as config.mml to the export/uk9/ directory:

```
gw mml> prov-exp:dirname=uk9
H323 Signaling Interface Sun Jan 7 14:29:11 2001
M SUCC
Successfully exported "active" to export/uk9/config.mml
```

The UNIX command shown in the following example executes MML in batch mode and imports the configuration file that was exported in the previous example:

```
mml> -b /opt/GoldWing/currentGW/export/uk9/config.mml
```

Related Commands

Command	Description
prov-add	Adds a component.
prov-cpy	Activates the configuration settings in the current provisioning session.
prov-dlt	Deletes a provisioned component.
prov-ed	Modifies a provisioned component.
prov-rtrv	Retrieves information about an existing provisioning session.
prov-sta	Establishes a provisioning session.
prov-stp	Terminates either a specified provisioning session or the current provisioning session.

prov-rtrv

The **prov-rtrv** command retrieves information about an existing provisioning session.

prov-rtrv:name=*MML name*

prov-rtrv:all

prov-rtrv:session

prov-rtrv:list

Note The **prov-rtrv:list** command is the only **prov-rtrv** command that can be executed outside of a provisioning session. Use the **prov-sta** command to start a provisioning command.

Syntax Description

name	The MML name for the elements that you want to display.
MML name	The MML component name for the component you want to display. Valid MML names are: • sys_config_static • sys_config_dynamic • h323_sys • ras • h245 • q931
all	Displays all components that have been provisioned.
session	Displays information about the provisioning session.
list	Provides a list of possible session names that you can use as the srcver parameter to prov-sta:srcver=uk9,dstver=uk10.

Usage Guidelines

See Chapter 3, "Provisioning the Cisco HSI," for a list of MML names, parameter names and their associated values.

Examples

The **prov-rtrv** command shown in the following example displays information about the MML name ras:

```
gw mml> prov-rtrv:name=ras
H323 Signaling Interface Sun Jan 7 14:46:01 2001
M SUCC
MML Name : ras.
Parameter: maxFail.
Value : 33.
```

The MML command shown in the following example displays information about the MML session:

```
gw mml> prov-rtrv:session
H323 Signaling Interface Sun Jan 7 14:46:01 2001
M RTRV
Session ID = mml 6 | davek
SRCVER = uk9
DSTVER = inter

gw mml> prov-rtrv:list
H323 Signaling Interface Sun Jan 7 14:46:01 2001
M RTRV

The following provisioning sessions are available:
uk9 matt inter
gw mml>
```

Related Commands

Command	Description
prov-add	Adds a component.
prov-cpy	Activates the configuration settings in the current provisioning session.
prov-dlt	Deletes a provisioned component.
prov-ed	Modifies a provisioned component.
prov-exp	Exports the current configuration of the Cisco HSI in MML command form to a file or files.
prov-sta	Establishes a provisioning session.
prov-stp	Terminates either a specified provisioning session or the current provisioning session.

prov-sta

The **prov-sta** command establishes a provisioning session. The data files are copied from the source version to the destination version.

prov-sta::srcver=*version***,dstver=***version*

Syntax Description

srcver=*version*	Selects a specific configuration version as the source for configuration changes. The srcver variable represents a directory that exists in $GWBASE/var/prov/. In place of the configuration version, you can also enter: • new—Specifies a new default session configuration; no existing source configuration is used. • active—Selects the active configuration as the source for configuration changes.
dstver=*version*	Specifies the output version directory for the configuration session results. The dstver variable represents a directory stored in $GWBASE/var/prov/.

Usage Guidelines

If the source and destination filenames are the same, the new configuration overwrites the old configuration. It is a good practice to copy an existing configuration instead of overwriting it so that you can return to a known configuration if there are problems with the new one.

If the source provisioning session has not been verified, the message “Note: This provisioning session has not been verified” is displayed, but the session starts normally.

If you try to start with a provisioning session that does not exist, an error is displayed, along with a list of available sessions.

Examples

The MML command in the following example starts a provisioning session named **nowt** and creates a new configuration named **blah**:

```
gw mml> prov-sta::srcver=nowt,dstver=blah
H323 Signaling Interface Sun Jan 7 13:32:07 2000
M DENY
The provisioning session called "nowt" does not exist.
The following configurations are available:
sanfran2 uk9 final
telco mgcpvia miki
transit dave matt
```

The MML command in the following example starts a provisioning session and creates a new configuration named **ver1**:

```
gw mml> PROV-STA::SRCVER="new", DSTVER="ver1"
H323 Signaling Interface Sun Jan 7 13:32:07 2001
M SUCC
Successfully started provisioning session "ver1" from "new".
```

The MML command in the following example starts a provisioning session, opens the existing configuration named **ver1**, and overwrites that configuration:

```
gw mml> PROV-STA::SRCVER="ver1", DSTVER="ver1"
```

The MML command shown in the following example starts a provisioning session, opens the existing configuration named **ver1**, and saves the updated configuration as **ver2**:

```
gw mml> PROV-STA::SRCVER="ver1", DSTVER="ver2"
```

Related Commands

Command	Description
prov-add	Adds a component.
prov-cpy	Activates the configuration settings in the current provisioning session.
prov-dlt	Deletes a provisioned component.
prov-ed	Modifies a provisioned component.
prov-exp	Exports the current configuration of the Cisco HSI in MML command form to a file or files.
prov-rtrv	Retrieves information about an existing provisioning session.
prov-stp	Terminates either a specified provisioning session or the current provisioning session.

prov-stp

The **prov-stp** command terminates the provisioning session and saves the configuration.

prov-stp:confirm

prov-stp:session name:confirm

Syntax Description

session name	Use the **rtrv-mml** command to retrieve the MML name given to the MML process that started the provisioning session.
confirm	If no confirm option is entered, the command is rejected and a message notifies you of the potential performance impact of this command.

Usage Guidelines

You can use the name given to an MML session to stop a provisioning session. Each MML session (not Telnet) gets an MML name: for example, mml1 or mml2. The maximum number of allowable MML sessions is 12.

If you log in to the Cisco HSI from an MML session and start a provisioning session (for example, gw mml> **prov-sta:srcver=new,dstver=uk9**), you can use the MML name (for example, mml2) to stop the session with **prov-stp** (for example, **prov-stp:mml2:confirm**).

Use the **rtrv-mml** command to display all active MML sessions (see rtrv-mml, page A-29).

Examples

The MML command in the following example terminates the current provisioning session:

```
gw mml> prov-stp:confirm
H323 Signaling Interface Sun Jan 7 14:46:01 2001
M SUCC
Successfully stopped provisioning session "ver1"
```

The MML command in the following example terminates the uk9 provisioning session:

```
gw mml> prov-stp:uk9:confirm
H323 Signaling Interface Day Mon 1 hh:mm:ss YYYY
M SUCC
Successfully stopped provisioning session "uk9"
gw mml>
```

If the previous session starts from an MML process assigned the name mml2, you can use the following MML command:

```
gw mml> prov-stp:mml2:confirm
H323 Signaling Interface Day Mon 1 hh:mm:ss YYYY
M SUCC
Successfully stopped provisioning session "uk9"
gw mml>
```

Related Commands

Command	Description
prov-add	Adds a component.
prov-cpy	Activates the configuration settings in the current provisioning session.
prov-dlt	Deletes a provisioned component.
prov-ed	Modifies a provisioned component.
prov-exp	Exports the current configuration of the Cisco HSI in MML command form to a file or files.
prov-rtrv	Retrieves information about an existing provisioning session.
prov-sta	Establishes a provisioning session.

quit

The **quit** command ends an MML session.

quit

Syntax Description

This command has no arguments or keywords.

Examples

The command in the following example ends an MML session.

```
gw mml> quit
```

radlog

The **radlog** command starts or stops RADVision logging into a specified log file.

radlog:[*file name***]:start | stop**

Caution This command is processor intensive and results in very large log files. Use this command only to retrieve information for single test calls, and do not use it on a live network that is processing numerous calls.

Syntax Description.

file name	The user-defined name of the log file. The actual filename has a *.log* suffix. The file is located in the logging directory defined in the configuration data (see Chapter 3, "Provisioning the Cisco HSI").

Examples

The command in the following example starts logging into a diagnostics log named file1:

```
gw mml> radlog:file1:start
H323 Signaling Interface Day Mon 1 hh:mm:ss YYYY
M SUCC
RADLogging requested to start
```

The following command logs RADVision to the standard log file:

```
gw mml> radlog::start
H323 Signaling Interface Day Mon 1 hh:mm:ss YYYY
M SUCC
RADLogging to standard log file
```

Related Commands

Command	Description
diaglog	Starts or stops event logging into a diagnostics log.
rtrv-log	Displays the logging level of a package or all packages.
set-log	Sets the logging level of a package or all packages.

restart-softw

The **restart-softw** command restarts the call processing application. It applies the provisioning data specified in the configVersion (if present) that overrides the existing active provisioning data.

restart-softw[*:configVersion***][***:confirm***]**

Syntax Description

configVersion	In *configVersion*, **init** is a keyword, and this command restarts the call processing application applying the etc/GWmain.conf configuration file as the provisioning data. If configVersion is an unverified provisioning session, the command fails.
confirm	If there are active calls, a notification is sent to the craft, and the command must be reentered with the *confirm* parameter to take effect.

Examples

In the following example, the call processing application restarts using the etc/GWmain.conf configuration files as the provisioning data:

```
gw mml> restart-softw:init
H323 Signaling Interface Day Mon 1 hh:mm:ss YYYY
M SUCC

Application is now restarting using the default provisioning session.
There are no active calls.
New call requests are rejected.
Call Processing now stopped.
Application will restart in 60 seconds
```

In the following example, the application would restart using the active provisioning session. There are no active calls, new call requests are rejected, and call processing is now stopped. The application is set to restart in 12 seconds.

```
gw mml> restart-softw
H323 Signaling Interface Day Mon 1 hh:mm:ss YYYY
M SUCC
```

In the following example, a restart passes an unverified provisioning session. The command fails. You cannot use an unverified provisioning session.

```
gw mml> restart-softw:config2
H323 Signaling Interface Day Mon 1 hh:mm:ss YYYY
M DENY
```

In the following example, a restart uses a specified verified provisioning session. The application restarts by using original as the provisioning session. There are no active calls, new call requests are rejected, and call processing is now stopped. The application is set to restart in 12 seconds.

```
gw mml> restart-softw:original
H323 Signaling Interface Day Mon 1 hh:mm:ss YYYY
M SUCC
```

rtrv-alms

The **rtrv-alms** command retrieves all active alarms.

rtrv-alms

rtrv-alms:cont

Syntax Description

cont	This parameter displays alarm events until you press **Ctrl-C**. All active alarms are displayed, and then a message appears (for example: "/* Listening for alarm events . . . (Ctrl-C to stop) */").

Examples

In the following example, the output contains the standard alarm definition and also a NACK or an ACK for noninformational alarms. NACK and ACK indicate the acknowledgement status of the alarm.

```
gw mml> rtrv-alms
GW Signaling Interface    2000-12-05 14:19:22
M   RTRV
"H323-GW1: 2000-11-27 11:25:12.259, ** ALM=\"VSC FAILURE\",SEV=MJ" NACK
"H323-GW1: 2000-11-27 11:25:13.260, ** ALM=\"CONFIGURATION FAILURE\",SEV=MJ" ACK
"H323-GW1: 2000-11-27 11:25:14.011, A^ ALM=\"ENDPOINT CHANNEL INTERFACE FAILURE\",SEV=IF"
"H323-GW1: 2000-11-27 11:25:14.012, A^ ALM=\"ENDPOINT CHANNEL INTERFACE FAILURE\",SEV=IF"
```

In the following example, the output displays alarm events until you press **Ctrl-C**:

```
gw mml> rtrv-alms:cont
GW Signaling Interface    2000-12-05 14:19:22
M   RTRV
"H323-GW1: 2000-11-27 11:25:12.259, ** ALM=\"VSC FAILURE\",SEV=MJ"
"H323-GW1: 2000-11-27 11:25:13.259,    ALM=\"VSC FAILURE\",SEV=MJ" STATE=CLEARED
"H323-GW1: 2000-11-27 11:25:13.260, ** ALM=\"CONFIGURATION FAILURE\",SEV=MJ"
"H323-GW1: 2000-11-27 11:25:14.011, A^ ALM=\"ENDPOINT CHANNEL INTERFACE FAILURE\",SEV=IF"
"H323-GW1: 2000-11-27 11:25:14.012, A^ ALM=\"ENDPOINT CHANNEL INTERFACE FAILURE\",SEV=IF"

/* Listening for alarm events... (Ctrl-C to stop) */

"H323-GW1: 2000-11-27 11:25:13.259, ** ALM=\"VSC FAILURE\",SEV=MJ"

/* Ctrl-C pressed */
```

Related Commands

Command	Description
ack-alm	Acknowledges that an alarm event is recognized but does not clear the alarm.
clr-alm	Clears an alarm event.

rtrv-calls

The **rtrv-calls** command displays all actively connected calls. If the *time elapsed* parameter is provided (in units of minutes), calls display only if they exceed the specified time. The output includes the call direction, time connected, calling and called address, and call reference.

rtrv-calls*[:time elapsed]*

Syntax Description

time elapsed	If the time elapsed parameter is provided (in units of minutes), calls display only if they have exceeded the specified time.

Examples

In the following example, the command displays all actively connected calls:

```
gw mml> rtrv-calls
H323 Signaling Interface Day Mon 1 hh:mm:ss YYYY
M SUCC

CallId SrcAddr DestAddr StartTime
124 04161234567 0299598125 2000-11-27 11:25:13.259
```

rtrv-ctr

The **rtrv-ctr** command displays the measurements for a counter group.

rtrv-ctr:*counter group*

Syntax Description

counter group	The name of an MML counter group (RAS, Q931, or H245).

Examples

In the following example, the command displays the measurements for the **RAS** counter group:

```
gw mml> rtrv-ctr:RAS
GW Signaling Interface    2000-12-05 14:19:22
M   RTRV
"H323-GW1:GROUP=\"RAS\",NAME=\"GK DISC ATT TOT\",VAL=10"
"H323-GW1:GROUP=\"RAS\",NAME=\"GK REG ATT TOT\",VAL=0"
"H323-GW1:GROUP=\"RAS\",NAME=\"GK REG SUCC TOT\",VAL=12"
"H323-GW1:GROUP=\"RAS\",NAME=\"GK RCV UNR ATT TOT\",VAL=100"
"H323-GW1:GROUP=\"RAS\",NAME=\"GK XMIT UNR SUCC TOT \",VAL=2000"
"H323-GW1:GROUP=\"RAS\",NAME=\"GK XMIT UNR ATT TOT\",VAL=20"
"H323-GW1:GROUP=\"RAS\",NAME=\"GK RCV UNR SUCC TOT\",VAL=10"
"H323-GW1:GROUP=\"RAS\",NAME=\""GK RLS ATT TOT\",VAL=20"
"H323-GW1:GROUP=\"RAS\",NAME=\"GK RLS SUCC TOT\",VAL=30"
"H323-GW1:GROUP=\"RAS\",NAME=\"GK INFO REPORT TOT\",VAL=40"
```

Related Commands

Command	Description
clr-meas	Resets a measurement counter.

rtrv-dest

The **rtrv-dest** command retrieves status information about the IP links and EISUP signalling path to the PGW 2200.

The output produced by this command shows the states of the H323 signalling gateway external interfaces to the PGW 2200s:

- IS: In Service
- OOS: Out Of Service
- MAN: Manual
- REM: Remote
- FLT: Fault
- NCFG: Not Configured

Examples

The MML command in the following example retrieves status information about the IP links and EISUP signalling path to the PGW 2200:

```
gw mml> rtrv-dest
   H323 Signalling Gateway   Thu Aug 23 01:15:32 2007
M  SUCC

        VscA:           ACT
        ipLink1toVscA:  IS
        ipLink2toVscA:  OOS NCFG

        VscB:           OOS NCFG
        ipLink1toVscB:  OOS NCFG
        ipLink2toVscB:  OOS NCFG

        EisupPath:      IS
gw mml>
```

Related Commands

Command	Description
set-dest-state	Changes the service state of an IP link or E-ISUP signaling path to in service (IS) or out of service (OOS).

rtrv-gapping

The **rtrv-gapping** command retrieves information about overload-triggered call gapping.

The following information displays:

- The active or inactive status of call gapping
- The percentage of calls that are gapped
- The type of calls to which gapping is applied

rtrv-gapping

Syntax Description

This command has no arguments or keywords

Examples

The following MML command retrieves the current levels of call gapping for all gapping clients:

```
gw mml> rtrv-gapping

Client Name      Direction        Level            Call Type        Active

Overload         Outgoing         10               Normal           No
Overload         Incoming         10               Normal           No
MML              Outgoing         20               All              Yes
MML              Incoming         30               All              Yes
```

Related Commands

Command	Description
set-gapping	Sets the type of calls to be gapped.

rtrv-log

The **rtrv-log** command displays the logging level of a package or all packages.

rtrv-log:package=*x*

rtrv-log:all

Syntax Description

package=*x*	Displays the logging level for the various packages that make up the Cisco HSI. For package names, see the "Log Message Packages" section on page 4-10.
all	Displays the logging levels of all packages.

Examples

In the following example, the command displays the logging levels of all packages:

```
gw mml> rtrv-log:all
H323 Signaling Interface Thu Dec 14 16:28:44 2000
M RTRV

Logging levels:
Application........0x0000
CallControl........0xFFFF
Connection.........0x0000
DataManager........0x0000
Eisup..............0xFFFF
FaultManager.......0x0000
Gapping............0x0000
H323...............0xFFFF
Infrastructure.....0x0000
OverLoad...........0x0000
ProcessManager.....0x0000
Provisioning.......0x0000
Signal.............0x0000
Snmp...............0x0000
SnmpSubagent.......0x0000
Statistics.........0x0000
Trace..............0x0000
UserInterface.......0x0000
```

Related Commands

Command	Description
diaglog	Starts and stops event logging into a diagnostics log.
radlog	Starts and stops RADVision logging into a specified log file.
set-log	Sets the logging level of a package or all packages.

rtrv-mml

The **rtrv-mml** command displays the following information:

- All active MML sessions
- Session numbers of all active MML sessions
- User IDs of the session originators

rtrv-mml

Syntax Description

This command has no arguments or keywords.

Examples In the following example, the command displays all active MML sessions, their sessions numbers, and the user IDs of the session originators.

```
gw mml> rtrv-mml

VSC H-323 Signaling Interface - H323-GW1 2000-06-20 10:04:28
M RTRV
mml1:matthewl
mml2:davek
```

rtrv-ne

The **rtrv-ne** command displays the type, hardware platform, vendor, location, version, and status of the Cisco HSI.

rtrv-ne

Syntax Description This command has no arguments or keywords.

Examples In the following example, the command displays the type, hardware platform, vendor, location, version, and status of the Cisco HSI.

```
gw mml> rtrv-ne

H323 Signaling Interface Thu Dec 14 16:29:19 2000
M RTRV

Type: H323 Signaling Interface
Hardware platform: Sun netra t1
Vendor: Cisco Systems, Inc.
Location: H323 - GW1
Version: R1_1_0
Platform Status:
Signaling interface: Active
Call processing: Running
```

rtrv-ne-health

The **rtrv-ne-health** command displays the following information about the Cisco HSI status:

- CPU load
- Disk space
- Number of currently connected calls
- Number of calls in setup

rtrv-ne-health

Syntax Description This command has no arguments or keywords.

Examples In the following example, the command displays information about the Cisco HSI status:

```
gw mml> rtrv-ne-health

VSC H-323 Signaling Interface - H323-GW1 2000-06-20 10:04:28
M RTRV

CPU Load:                  23%
Disk space:                123456
Number of connected calls: 23
Number of calls in setup:  12
```

rtrv-overload

The **rtrv-overload** command displays overload status information and value settings for the three provisionable levels of overload.

rtrv-overload

Syntax Description This command has no arguments or keywords.

Examples In the following example, the command displays overload status information:

```
gw mml> rtrv-overload
H323 Signaling Interface Tue Jan 30 11:21:45 2001
M SUCC
Overload/Gapping Information
NumCalls : 0 | CPU : 7% | DiskUsage : 27%
Status : Not in Ovld
Overload Configuration
DiskUsageLimit : 29%
OvldSampleRate : 3000ms
OvldLevel1Percent : 65%
OvldLevel1Filter : NORMAL
OvldLevel1ThreshLowerCpu : 30%
OvldLevel1ThreshUpperCpu : 35%
OvldLevel1ThreshLowerCalls : 800
OvldLevel1ThreshUpperCalls : 1000
OvldLevel2Percent : 75%
OvldLevel2Filter : ALL
OvldLevel2ThreshLowerCpu : 45%
OvldLevel2ThreshUpperCpu : 50%
OvldLevel2ThreshLowerCalls : 1100
OvldLevel2ThreshUpperCalls : 1400
OvldLevel3Percent : 90%
OvldLevel3Filter : NORMAL
OvldLevel3ThreshLowerCpu : 55%
OvldLevel3ThreshUpperCpu : 65%
OvldLevel3ThreshLowerCalls : 1400
OvldLevel3ThreshUpperCalls : 1600
```

Related Commands

Command	Description
set-overload	Defines the overload handling criteria and behavior.

rtrv-softw

The **rtrv-softw** command displays the status of the Cisco HSI and call processing activity. The following software states can be displayed for the Cisco HSI:

- Not running
- Starting
- Active
- Restart pending
- Halt pending
- Reboot pending

The following software states can be displayed for call processing:

- Running
- Idle pending
- Idle

rtrv-softw

Note When the Cisco HSI is in the restart pending, halt pending, or reboot pending software state, the **sta-callproc** command cancels the pending state.

Syntax Description

This command has no arguments or keywords.

Examples

In the following example, the command displays the status of the Cisco HSI and call processing activity:

```
gw mml> rtrv-softw
VSC H-323 Signaling Interface - H323-GW1 2000-06-20 10:04:28
M RTRV
Platform Status:
Signaling interface:  Active
Call processing:     Running
```

set-dest-state

The **set-dest-state** command changes the service state of an IP link or E-ISUP signaling path to IS (in service) or OOS (out of service).

set-dest-state:ipLink1toVscA:IS|OOS

set-dest-state:ipLink2toVscA:IS|OOS

set-dest-state:ipLink1toVscB:IS|OOS

set-dest-state:ipLink2toVscB:IS|OOS

set-dest-state:EisupPath:IS|OOS

set-dest-state:ipLinkNms:IS|OOS

Syntax Description

IS	In service.
OOS	Out of service.

Examples

In the following example, the command changes the service state of an IP link signaling path to IS:

```
gw mml> set-dest-state:ipLink1toVscA:state=IS
H323 Signaling Interface Day Mon 1 hh:mm:ss YYYY
M SUCC

Initiating state change of ipLink1toVscA to IS
gw mml> set-dest-state:ipLink1toVscA:state=OOS
H323 Signaling Interface Day Mon 1 hh:mm:ss YYYY
M SUCC

Initiating state change of ipLink1toVscA to OOS
```

Related Commands

Command	Description
rtrv-dest	Retrieves status information about the IP links and E-ISUP signaling path to the PGW 2200.

set-gapping

The **set-gapping** command sets the type of calls to be gapped.

set-gapping: inc | otg | both : calltype=normal | all, percent=*number*

Syntax Description

inc	Gaps calls from the H.323 network.
otg	Gaps calls from the PSTN over E-ISUP.
both	Gaps calls originating from either side.
normal	Gaps all calls except priority and emergency calls.
all	Gaps calls of all types.
number	Specifies the percentage of calls rejected due to call gapping.

Usage Guidelines

If call gapping is set to 100 percent, all calls are gapped irrespective of the normal or all parameter setting.

If the overload condition is active and call gapping is active, the higher of the two percentage values determines whether new call attempts are accepted or rejected.

Examples

In the following example, the command sets all calls to be gapped and specifies that 50 percent of the calls be rejected due to call gapping:

```
gw mml> set-gapping:both:calltype=all,percent=50
H323 Signaling Interface Day Mon 1 hh:mm:ss YYYY
M SUCC
Successfully set gapping for target 'both', calltype 'all', and percentage 50
```

Related Commands

Command	Description
rtrv-gapping	Retrieves information about overload-triggered call gapping.

set-log

The **set-log** command sets the logging level of a package or all packages.

set-log:package:level=*level,* **[confirm]**

set-log:all:level=*level***, [confirm]**

Syntax Description

package	One of the packages in the Cisco HSI. For a list of package names, see the "Log Message Packages" section on page 4-10.
level	Logging levels are set through the use of hexadecimal numbers between 0x0000 and 0xFFFF. The higher the number, the higher the level of debug.
confirm	If any client level parsing fails on the data session, a **confirm** is needed for the data to be activated.

Usage Guidelines

Logging at any level implies that upper levels are included. When you are setting logging with the level DEBUG, a confirmation is required because the amount of data logged affects service. For a list of the packages that can log messages, see the "Log Message Packages" section on page 4-10.

Examples

In the following example, the command sets the logging level of the package gapping to 0xFFE0:

```
gw mml> set-log:gapping:0xFFE0
M SUCC
logging level for package gapping set to 0xFFE0
```

Related Commands

Command	Description
diaglog	Starts and stops event logging into a diagnostics log.
radlog	Starts and stops RADVision logging into a specified log file.
rtrv-log	Displays the logging level of a package or all packages.

set-overload

The **set-overload** command defines the overload handling criteria and behavior.

set-overload: level1|level2|level3:cpu,lower=*number*, **upper=***number*

set-overload: level1|level2|level3:calls,lower=*number*, **upper=***number*

set-overload: level1|level2|level3:gap,normal|all : *number*

Syntax Description

level 1 | level 2 | level 3	Overload behavior can be provisioned at three separate levels: 1, 2, and 3 (rising in severity).
lower=*number*	The lower threshold for overload detection and restoration of normal call handling service.
upper=*number*	The upper threshold for overload detection and restoration of normal call handling service.

Usage Guidelines

The **set-overload** command defines the upper and lower thresholds for overload detection and restoration of normal call handling service. The percentage of calls to be gapped and the type of calls to be gapped can also be configured. Any changes made become active immediately.

The lower value must always be less than the upper value. If the call gap percentage is set to 0, the system takes no recovery action when overload is encountered, but the appropriate alarm is raised.

Inconsistent threshold settings for different levels can destabilize call processing. For successful execution of this command, ensure that threshold settings are consistent, as follows:

- The number of calls gapped at level 2 must be greater than or equal to the number of calls gapped at level 1.
- The number of calls gapped at level 3 must be greater than or equal to the number of calls gapped at levels 1 and 2.
- The lower level value of CPU occupancy must always be less than the upper level value.
- The lower level value of CPU occupancy and the number of calls for level 2 must be greater than or equal to the corresponding values for level 1.
- The lower level value of CPU occupancy and the number of calls for level 3 must be greater than or equal to the corresponding values for levels 1 and 2.

Examples

In the following example, the command sets the overload handling criteria and behavior at level 1, sets the cpu to the lower threshold of 10, and sets the upper threshold to 14.

```
gw mml> set-overload:level1:cpu, lower =10, upper = 14
H323 Signaling Interface Day Mon 1 11:21:28 2001
M SUCC
Successfully added that configuration item.
```

Related Commands

Command	Retrieve
rtrv-overload	Displays the overload status and the data values for the three provisionable levels of overload

sta-callproc

The **sta-callproc** command starts call processing.

sta-callproc

Syntax Description

This command has no arguments or keywords.

Examples

In the following example, the command starts call processing:

```
gw mml> sta-softw
gw mml> sta-callproc

H323 Signaling Interface Thu Dec 14 16:31:09 2000
M SUCC

Starting call processing.
```

Related Commands

Command	Description
sta-softw	Starts the call processing application.

sta-softw

The **sta-softw** command starts the call processing application.

sta-softw

Syntax Description

This command has no arguments or keywords.

Examples

In the following example, the command starts the call processing application:

```
gw mml> sta-softw

H323 Signaling Interface Day Mon 1 hh:mm:ss YYYY
M SUCC

The Call Processing Application is starting.
```

Related Commands

Command	Description
sta-callproc	Starts call processing

sta-trc

The **sta-trc** command starts the call processing tracing function.

sta-trc:Calltype=*calltype*,**TraceLevel=***trace level*[,**CdAddress=***address*]

[,**CgAddress=***address*],**log=***filename*[,**prd=*n***]

Syntax Description

Calltype=*calltype*	This parameter defines the type of call to be traced. The possible values are: • inc—For H.323-originated calls • otg—For E-ISUP-originated calls • both—For calls originating from either side
TraceLevel=*trace level*	This parameter defines the level of detail that is recorded in the call trace. The possible values are: • terse—Traces only incoming/outgoing message names and RADvision API calls. • brief—In addition to terse, traces internal interfaces, and call state changes. • verbose—Traces all messages and their contents, RADvision API calls and their contents, internal interfaces, and call state changes.
CdAddress=*address*	A filter used to trace calls using only the specified leading digits within the called address. A match is performed on these digits and the initial called address digits contained within the E-ISUP IAM message or the H225 SETUP message.
CgAddress=*address*	A filter used to trace calls using only the specified leading digits within the calling address. A match is performed on these digits and the initial calling address digits contained within the E-ISUP IAM message or the H.225 SETUP message.
log=*filename*	The filename for the trace output.
prd=*n*	The trace period (in seconds). At expiration of this period, the trace log is closed. If no value is specified, the period defaults to 1800 seconds (30 minutes).

Defaults

The default trace for all calls is 30 minutes.

Usage Guidelines

Only one trace command at a time can be active. If an additional request is issued, the command is rejected with a call trace already active indication.

Examples

In the following example, the command starts the call processing tracing function:

```
gw mml> sta-trc:Calltype=both, TraceLevel=terse, dAddress=012,CgAddress=013, log=tlog.txt,
prd=10

H323 Signaling Interface Day Mon 1 hh:mm:ss YYYY
M SUCC
```

Related Commands

Command	Description
stp-trc	Halts the tracing currently active and closes the trace file.

stp-call

The **stp-call** command terminates a currently active call by forcing a release of the call. Disconnect/release messages are sent in both directions.

stp-call:callref=x

stp-call:all

Syntax Description

callref	Refers to a positive integer.
all	Stops all calls.

Examples

In the following example, the command terminates a currently active call by forcing a release of the call with a callref of 33.

```
gw mml> stp-call:callref=33

H323 Signaling Interface Thu Dec 14 16:43:54 2000
M SUCC

Stopped call 33
```

Related Commands

Command	Description
rtrv-calls	Displays all actively connected calls.

stp-callproc

The **stp-callproc** command stops further call processing by immediately terminating the handling of new call requests.

stp-callproc[*:timeout=T***]**

Syntax Description

timeout	If no timeout period is provided, existing calls are released immediately. If a timeout period is provided, existing calls are released after the specified amount of time has elapsed. When all calls have been released, a notification message is sent to the craft terminal.
T	T is in seconds.

Examples

In the following example, the command stops further call processing by immediately terminating the handling of new call requests:

```
gw mml> stp-callproc

H323 Signaling Interface Thu Dec 14 16:27:07 2000
M SUCC

Stopped accepting new calls. Existing calls will be released in 5 seconds.

Stopping Call Processing.
```

stp-softw

The **stp-softw** command stops the call processing application. This command causes the Cisco HSI to terminate.

stp-softw[:confirm]

Syntax Description

confirm	If there are active calls, a notification is sent to the craft. In order for the command to take effect, it must be reentered with the confirm parameter.

Examples

In the following example, the command stops the call processing application:

```
gw mml> stp-softw

H323 Signaling Interface Thu Dec 14 16:27:36 2000
M SUCC

There are no active calls.
Application is now stopping
```

stp-trc

The **stp-trc** command halts the currently active tracing and closes the trace file.

stp-trc

Syntax Description

This command has no arguments or keywords

Examples

In the following example, the command halts the currently active tracing and closes the trace file:

```
gw mml> stp-trc
H323 Signaling Interface Day Mon 1 hh:mm:ss YYYY
M SUCC
Tracing deactivated.
```

Related Commands

Command	Description
sta-trc	Starts the call processing tracing function.

Skeleton Configuration File

Provisionable and configurable data elements within the Cisco H.323 Signaling Interface (HSI) are grouped as dynamic, static, or constant data. You can modify dynamic and static data through the provisioning MML commands.

At startup, the Cisco HSI reads a skeleton file, which contains information for every configurable item within the system. It determines which group the item is in and what its possible ranges are. The format for each item is:

PackageName ItemName GroupValueRange

The package name is an MML name for dynamic and static data, and service package names (determined within the detailed design) for constant data.

The item name is as specified in the detailed design (that is, all constant data).

The group is one of the following:

- S = Static data
- D = Dynamic data
- C = Constant data

The ValueRange is one of the following:

- [ ] = Number: No range
- [x–y] = Number: Acceptable range *x* to *y*
- STRING = String value
- NULL = No associated value

An example of an MML command follows:

```
RASmaxFailD[1-200]
```

In the preceding example, the PackageName is RAS, and the ItemName is maxFail. It is a dynamic configuration item, and it is a numerical data type with an acceptable range of 1 to 200.

The skeleton file allows the system technician to specify whether data is provisionable and whether provisionable data is static or dynamic.

Note The skeleton configuration file should be modified only by authorized personnel.

The following is an example of a skeleton file:

```
#******  sys_config_static ************************
#Static
SYS_CONFIG_STATIC          Nodeid                          S          STRING
SYS_CONFIG_STATIC          HOST_PORT_NUMBER1       S          [0-65535]
SYS_CONFIG_STATIC          HOST_PORT_NUMBER2       S          [0-65535]
SYS_CONFIG_STATIC          HOST_IPADDR1            S          STRING
SYS_CONFIG_STATIC          HOST_IPADDR2            S          STRING
SYS_CONFIG_STATIC          VSCA_IPADDR1            S          STRING
SYS_CONFIG_STATIC          VSCA_IPADDR2            S          STRING
SYS_CONFIG_STATIC          VSCB_IPADDR1            S          STRING
SYS_CONFIG_STATIC          VSCB_IPADDR2            S          STRING
SYS_CONFIG_STATIC          VSCA_PORT_NUMBER1       S          [0-65535]
SYS_CONFIG_STATIC          VSCA_PORT_NUMBER2       S          [0-65535]
SYS_CONFIG_STATIC          VSCB_PORT_NUMBER1       S          [0-65535]
SYS_CONFIG_STATIC          VSCB_PORT_NUMBER2       S          [0-65535]

SYS_CONFIG_STATIC          ClipClirSupported       S          STRING
SYS_CONFIG_STATIC          DualCLISupported        S          STRING
SYS_CONFIG_STATIC          RaiSupported            S          STRING
SYS_CONFIG_STATIC          DtmfSupportedDirection  S          STRING
SYS_CONFIG_STATIC          DtmfSupportedType       S          STRING
SYS_CONFIG_STATIC          NetchatSupported        S          STRING
SYS_CONFIG_STATIC          H225PavoSupported       S          STRING
SYS_CONFIG_STATIC          PavoRedirScreeningInd   S          [0-3]
SYS_CONFIG_STATIC          PavoRedirReason         S          [0-15]
SYS_CONFIG_STATIC          PavoRedirPresInd        S          [0-3]
SYS_CONFIG_STATIC          CliInDisplaySupported   S          STRING
SYS_CONFIG_STATIC          T38MaxVal               S          STRING
SYS_CONFIG_STATIC          T38Options              S          STRING
SYS_CONFIG_STATIC          AsymmetricHandlingSupported  S     STRING
SYS_CONFIG_STATIC          UseConfID               S          STRING

SYS_CONFIG_STATIC          CC_EC_DEFAULT           S          STRING
SYS_CONFIG_STATIC          CC_HC_DEFAULT           S          STRING
SYS_CONFIG_STATIC          CC_EC_AccessBarred      S          STRING
SYS_CONFIG_STATIC          CC_EC_Acknowledgement   S          STRING
SYS_CONFIG_STATIC          CC_EC_AddressIncomplete S          STRING
SYS_CONFIG_STATIC          CC_EC_AnonymousCallRejection  S    STRING
```

```
SYS_CONFIG_STATIC       CC_EC_BlacklistBNumberMatched S       STRING
SYS_CONFIG_STATIC       CC_EC_BlacklistCliLengthInvalid S     STRING
SYS_CONFIG_STATIC       CC_EC_BlacklistCliMatched S           STRING
SYS_CONFIG_STATIC       CC_EC_BlacklistCpcRestricted S        STRING
SYS_CONFIG_STATIC       CC_EC_BlacklistNoCli S                STRING
SYS_CONFIG_STATIC       CC_EC_BlacklistNoaRestricted S        STRING
SYS_CONFIG_STATIC       CC_EC_Busy S                          STRING
SYS_CONFIG_STATIC       CC_EC_CallRejectCallGapping S         STRING
SYS_CONFIG_STATIC       CC_EC_CallTerminated S                STRING
SYS_CONFIG_STATIC       CC_EC_CallTypeIncompatible S          STRING
SYS_CONFIG_STATIC       CC_EC_CallingDroppedWhileOnHold S     STRING
SYS_CONFIG_STATIC       CC_EC_CallingPartyOffHold S           STRING
SYS_CONFIG_STATIC       CC_EC_ChannelOutOfService S           STRING
SYS_CONFIG_STATIC       CC_EC_Congestion S                    STRING
SYS_CONFIG_STATIC       CC_EC_CotFailure S                    STRING
SYS_CONFIG_STATIC       CC_EC_CugAccessBarred S               STRING
SYS_CONFIG_STATIC       CC_EC_DteControlledNotReady S         STRING
SYS_CONFIG_STATIC       CC_EC_DteUncontrolledNotReady S       STRING
SYS_CONFIG_STATIC       CC_EC_ExcessiveDigCallProceeding S    STRING
SYS_CONFIG_STATIC       CC_EC_FacilityNotRegistered S         STRING
SYS_CONFIG_STATIC       CC_EC_FlowControlledCongestion S      STRING
SYS_CONFIG_STATIC       CC_EC_GroupRestrictions S             STRING
SYS_CONFIG_STATIC       CC_EC_IncomingCallsBarred S           STRING
SYS_CONFIG_STATIC       CC_EC_InterceptedSubscriber S         STRING
SYS_CONFIG_STATIC       CC_EC_InterworkUnspec S               STRING
SYS_CONFIG_STATIC       CC_EC_InvalidCallRef S                STRING
SYS_CONFIG_STATIC       CC_EC_MesgWithUnrecElemDiscarded S    STRING
SYS_CONFIG_STATIC     CC_EC_MessageNotUnderstood S            STRING
SYS_CONFIG_STATIC     CC_EC_MisroutedCallPortedNumber S       STRING
SYS_CONFIG_STATIC     CC_EC_NetworkAddressExtensionError S    STRING
SYS_CONFIG_STATIC     CC_EC_NetworkTermination S              STRING
SYS_CONFIG_STATIC     CC_EC_NewDestination S                  STRING
SYS_CONFIG_STATIC     CC_EC_NumberUnobtainable S              STRING
SYS_CONFIG_STATIC     CC_EC_OperatorPriorityAccess S          STRING
SYS_CONFIG_STATIC     CC_EC_OutOfCatchmentArea S              STRING
SYS_CONFIG_STATIC     CC_EC_OutgoingCallsBarred S             STRING
SYS_CONFIG_STATIC    CC_EC_PermanentIcb S                     STRING
SYS_CONFIG_STATIC    CC_EC_PortedNumber S                     STRING
```

SYS_CONFIG_STATIC	CC_EC_PreemptionCctUnavailable	S	STRING
SYS_CONFIG_STATIC	CC_EC_Prefix0DialledInError	S	STRING
SYS_CONFIG_STATIC	CC_EC_Prefix1DialledInError	S	STRING
SYS_CONFIG_STATIC	CC_EC_Prefix1NotDialled	S	STRING
SYS_CONFIG_STATIC	CC_EC_PriorityForcedRelease	S	STRING
SYS_CONFIG_STATIC	CC_EC_Proprietary	S	STRING
SYS_CONFIG_STATIC	CC_EC_ProtErrThresholdExceeded	S	STRING
SYS_CONFIG_STATIC	CC_EC_ProtocolErrorUnspec	S	STRING
SYS_CONFIG_STATIC	CC_EC_Reject	S	STRING
SYS_CONFIG_STATIC	CC_EC_RejectedDivertedCall	S	STRING
SYS_CONFIG_STATIC	CC_EC_RemoteProcError	S	STRING
SYS_CONFIG_STATIC	CC_EC_RepeatAttempt	S	STRING
SYS_CONFIG_STATIC	CC_EC_RouteOutOfService	S	STRING
SYS_CONFIG_STATIC	CC_EC_SelectiveCallBarring	S	STRING
SYS_CONFIG_STATIC	CC_EC_ServiceIncompatible	S	STRING
SYS_CONFIG_STATIC	CC_EC_ServiceTemporarilyUnavailable	S	STRING
SYS_CONFIG_STATIC	CC_EC_ServiceUnavailable	S	STRING
SYS_CONFIG_STATIC	CC_EC_SignalNotUnderstood	S	STRING
SYS_CONFIG_STATIC	CC_EC_SignalNotValid	S	STRING
SYS_CONFIG_STATIC	CC_EC_SignallingSystemIncompatible	S	STRING
SYS_CONFIG_STATIC	CC_EC_SubControlledIcb	S	STRING
SYS_CONFIG_STATIC	CC_EC_SubNotFoundDle	S	STRING
SYS_CONFIG_STATIC	CC_EC_SubscriberCallTerminate	S	STRING
SYS_CONFIG_STATIC	CC_EC_SubscriberIncompatible	S	STRING
SYS_CONFIG_STATIC	CC_EC_SubscriberMoved	S	STRING
SYS_CONFIG_STATIC	CC_EC_SubscriberOutOfService	S	STRING
SYS_CONFIG_STATIC	CC_EC_TemporaryOos	S	STRING
SYS_CONFIG_STATIC	CC_EC_TerminalCongestion	S	STRING
SYS_CONFIG_STATIC	CC_EC_Transferred	S	STRING
SYS_CONFIG_STATIC	CC_EC_TranslationOos	S	STRING
SYS_CONFIG_STATIC	CC_EC_UnallocatedDestNumber	S	STRING
SYS_CONFIG_STATIC	CC_EC_UndefinedBg	S	STRING
SYS_CONFIG_STATIC	CC_EC_Unknown	S	STRING
SYS_CONFIG_STATIC	CC_EC_UnrecElemPassedOn	S	STRING
SYS_CONFIG_STATIC	CC_EC_VacentCode	S	STRING
SYS_CONFIG_STATIC	CC_EC_WhitelistCliNotMatched	S	STRING

```
#****** system ************************
#Static
H323_SYS            cidAssociate          S          NULL
H323_SYS            vtNodeCount           S          [0-65535]
H323_SYS            channels              S          [0-65535]
H323_SYS            messages              S          [0-65535]
H323_SYS            pktChans              S          [0-65535]
H323_SYS            protocols             S          [0-65535]
H323_SYS            maxProcs              S          [0-65535]
H323_SYS            maxBuffSize           S          [0-65535]
H323_SYS            maxCalls              S          [0-5000]
H323_SYS            maxChannels           S          [0-65535]


#****** sys_config_dynamic*************
#Dynamic
SYS_CONFIG_DYNAMIC      AlternateGatekeeperIp          D          STRING
SYS_CONFIG_DYNAMIC      AlternateGatekeeperPort        D          [0-65535]
SYS_CONFIG_DYNAMIC      AlternateGatekeeperId          D          STRING
SYS_CONFIG_DYNAMIC      Logdirectory                   D          STRING
SYS_CONFIG_DYNAMIC      LogFilenamePrefix              D          STRING
SYS_CONFIG_DYNAMIC      LogPrio                        D          STRING
SYS_CONFIG_DYNAMIC      LogFileRotateSize              D          [1000-65535]
SYS_CONFIG_DYNAMIC      LogFileRotateInterval          D          [0-65535]
SYS_CONFIG_DYNAMIC      IPAddrRNMS                     D          STRING
SYS_CONFIG_DYNAMIC      DiskUsageLimit                 D          [20-100]
SYS_CONFIG_DYNAMIC      OvldSampleRate                 D          [500-10000]
SYS_CONFIG_DYNAMIC      OvldLevel1Percent              D          [0-100]
SYS_CONFIG_DYNAMIC      OvldLevel1Filter               D          STRING
SYS_CONFIG_DYNAMIC      OvldLevel1ThreshLowerCalls     D          [1-5000]
SYS_CONFIG_DYNAMIC      OvldLevel1ThreshUpperCalls     D          [1-5000]
SYS_CONFIG_DYNAMIC      OvldLevel1ThreshLowerCPU       D          [0-100]
SYS_CONFIG_DYNAMIC      OvldLevel1ThreshUpperCPU       D          [0-100]
SYS_CONFIG_DYNAMIC      OvldLevel2Percent              D          [0-100]
SYS_CONFIG_DYNAMIC      OvldLevel2Filter               D          STRING
SYS_CONFIG_DYNAMIC      OvldLevel2ThreshLowerCalls     D          [1-5000]
SYS_CONFIG_DYNAMIC      OvldLevel2ThreshUpperCalls     D          [1-5000]
SYS_CONFIG_DYNAMIC      OvldLevel2ThreshLowerCPU       D          [0-100]
```

```
SYS_CONFIG_DYNAMIC      OvldLevel2ThreshUpperCPU    D       [0-100]
SYS_CONFIG_DYNAMIC      OvldLevel3Percent           D       [0-100]
SYS_CONFIG_DYNAMIC      OvldLevel3Filter            D       STRING
SYS_CONFIG_DYNAMIC      OvldLevel3ThreshLowerCalls  D       [1-5000]
SYS_CONFIG_DYNAMIC      OvldLevel3ThreshUpperCalls  D       [1-5000]
SYS_CONFIG_DYNAMIC      OvldLevel3ThreshLowerCPU    D       [0-100]
SYS_CONFIG_DYNAMIC      OvldLevel3ThreshUpperCPU    D       [0-100]
SYS_CONFIG_DYNAMIC      CiAgentScanPeriod           D       []
SYS_CONFIG_DYNAMIC      AlarmDebounceTime           D       [0-60000]
SYS_CONFIG_DYNAMIC      RegFailureReleaseCause      D       [1-127]

#****** ras ************************
#Static
RAS         manualRAS                               S       NULL

#****** ras ************************
#Dynamic
RAS         responseTimeOut                         D       [1-200]
RAS         maxFail                                 D       [1-200]
RAS         allowCallsWhenNonReg                    D       NULL
RAS         manualRegistration                      D       NULL
RAS         timeToLive                              D       [1-65535]
RAS         rasPort                                 D       [0-65535]
RAS         compare15bitRasCrv                      D       NULL
RAS         maxRetries                              D       [1-200]
RAS         maxMulticastTTL                         D       [0-200]
RAS         preGrantedArqUse                        D       STRING
RAS             manualDiscovery.ipAddress           D       STRING
RAS             manualDiscovery.port                D       [0-65535]
RAS             gateway.prefix[i]                   D       STRING
RAS             gatekeeperId                        D       STRING
RAS             terminalAlias[i].e164               D       STRING
RAS             terminalAlias[i].h323ID             D       STRING
RAS             endpointVendor.t35CountryCode       D       [0-255]
RAS             endpointVendor.t35Extension         D       [0-255]
RAS             endpointVendor.manufacturerCode     D       [0-65535]
RAS             endpointVendor.productId            D       STRING
RAS             endpointVendor.versionId            C       STRING
```

```
#****** q931 ************************
#Dynamic
Q931        responseTimeOut                 D       [1-200]
Q931        connectTimeOut                  D       [1-20000]
Q931        callSignalingPort               D       [0-65535]
Q931        maxCalls                        D       [0-5000]
Q931        notEstablishControl             D       NULL
Q931        overlappedSending               C       NULL
Q931        earlyH245                       C       NULL
Q931        h245tunneling                   C       NULL

#****** h245 ************************
#Dynamic
H245        masterSlave.terminalType        D       [0-255]
H245        masterSlave.manualResponse      D       NULL
H245        masterSlave.timeout             D       [0-65535]
H245        masterSlave.manualOperation     D       NULL
H245        channelsTimeout                 D       [0-65535]
H245        roundTripTimeout                D       [0-65535]
H245        requestCloseTimeout             D       [0-65535]
H245        requestModeTimeout              D       [0-65535]
H245        mediaLoopTimeout                D       [0-65535]
H245        caps.manualOperation            D       NULL
H245        caps.timeout                    D       [0-65535]
H245        caps.maxAudioDelay              D       [0-1023]
H245        caps.table[i].entryNo           D       [1-65535]

H245        caps.table[i].audio.g711Alaw64k     D       [1-256]
H245        caps.table[i].audio.g711Alaw56k     D       [1-256]
H245        caps.table[i].audio.g711Ulaw64k     D       [1-256]
H245        caps.table[i].audio.g711Ulaw56k     D       [1-256]
H245        caps.table[i].audio.g722at64k       D       [1-256]
H245        caps.table[i].audio.g722at56k       D       [1-256]
H245        caps.table[i].audio.g722at48k       D       [1-256]
H245        caps.table[i].audio.g728            D       [1-256]
H245        caps.table[i].audio.g729            D       [1-256]
H245        caps.table[i].audio.g729.echo[i]22k D       [1-256]
```

```
H245        caps.table[i].audio.g7231.maxAudioFrames        D        [1-256]
H245        caps.table[i].audio.g7231.silenceSuppression    D        [1-256]

H245        chan[i].name                                    D        STRING
H245        chan[i].audio.g711Alaw64k                       D        [1-256]
H245        chan[i].audio.g711Alaw56k                       D        [1-256]
H245        chan[i].audio.g711Ulaw64k                       D        [1-256]
H245        chan[i].audio.g711Ulaw56k                       D        [1-256]
H245        chan[i].audio.g722at64k                         D        [1-256]
H245        chan[i].audio.g722at56k                         D        [1-256]
H245        chan[i].audio.g722at48k                         D        [1-256]
H245        chan[i].audio.g728                              D        [1-256]
H245        chan[i].audio.g729                              D        [1-256]
H245        chan[i].audio.g7231.maxAudioFrames              D        [1-256]
H245        chan[i].audio.g7231.silenceSuppression          D        [1-256]

H245        modes[i].name                                   D        STRING
H245        modes[i].audio.g711Alaw64k                      D        NULL
H245        modes[i].audio.g711Alaw56k                      D        NULL
H245        modes[i].audio.g711Ulaw64k                      D        NULL
H245        modes[i].audio.g711Ulaw56k                      D        NULL
H245        modes[i].audio.g722at64k                        D        NULL
H245        modes[i].audio.g722at56k                        D        NULL
H245        modes[i].audio.g722at48k                        D        NULL
H245        modes[i].audio.g728                             D        NULL
H245        modes[i].audio.g729                             D        NULL
H245        modes[i].audio.g7231                            D        NULL

#****************** From GWmain.static.conf ******************
#Logging
#Const

Logging     Application             D       [0-65535]
Logging     CallControl             D       [0-65535]
Logging     CC                      D       [0-65535]
Logging     Connection              D       [0-65535]
Logging     DataManager             D       [0-65535]
Logging     Eisup                   D       [0-65535]
```

```
Logging     FaultManager        D       [0-65535]
Logging     Gapping             D       [0-65535]
Logging     H323                D       [0-65535]
Logging     Infrastructure      D       [0-65535]
Logging     Overload            D       [0-65535]
Logging     ProcessManager      D       [0-65535]
Logging     Provisioning        D       [0-65535]
Logging     Signal              D       [0-65535]
Logging     Snmp                D       [0-65535]
Logging     SnmpSubagent        D       [0-65535]
Logging     Statistics          D       [0-65535]
Logging     Trace               D       [0-65535]
Logging     UserInterface       D       [0-65535]
Logging     Configuration       D       [0-65535]
Logging     Timer               D       [0-65535]
Logging     EISUP               D       [0-65535]
Logging     OTLogging           D       STRING


##
#### Call Control Package
##
CCPackage   Hash                        C               STRING
CCPackage   Pound                       C               STRING
CCPackage   Star                        C               STRING
CCPackage   StopDigit                   C               STRING

CCPackage   A_CC_ChargeInd              D               []
CCPackage   A_CC_tEndToEndMethod        C               []
CCPackage   A_CC_tLineUser              D               []
CCPackage   A_CC_tLineStatus            C               []
CCPackage   A_CC_MLC_Action             C               []
CCPackage   A_CC_tSCCPMethod            C               []
CCPackage   A_CC_Interworking           D               []
CCPackage   A_CC_tEndToEndInfAvail      C               []
CCPackage   A_CC_tIsdnAllTheWay         D               []
CCPackage   A_CC_tEchoCancIr            C               []
CCPackage   A_CC_tLineAccess            D               []
```

CCPackage	A_CC_BNumDataNOA	D	[]
CCPackage	A_CC_BNumDataNPI	C	[]
CCPackage	A_CC_BNumDataINN	D	[]
CCPackage	A_CC_ANumDataNOA	D	[]
CCPackage	A_CC_Clir	D	[]
CCPackage	A_CC_ANumDataSI	D	[]
CCPackage	A_CC_ANumDataNPI	C	[]
CCPackage	A_CC_A_Cli	C	[]
CCPackage	A_CC_AddANumDataNOA	C	[]
CCPackage	A_CC_AddANumDataSI	D	[]
CCPackage	A_CC_AddANumDataNPI	C	[]
CCPackage	A_CC_AddANumDataCLIR	D	[]
CCPackage	A_CC_oLinecall	D	[]
CCPackage	A_CC_Location	D	[]
CCPackage	A_CC_CodeStandard	C	[]
CCPackage	A_CC_ProgressRestrict	C	[]
CCPackage	A_CC_oIsdnPref	C	[]
CCPackage	A_CC_oIsdnAllTheWay	C	[]
CCPackage	A_CC_oEndToEndInfAvail	C	[]
CCPackage	A_CC_oNatInd	C	[]
CCPackage	A_CC_oLSPP	C	[]
CCPackage	A_CC_oNBit	C	[]
CCPackage	A_CC_oPORC	C	[]
CCPackage	A_CC_oPBit	C	[]
CCPackage	A_CC_oEndToEndMethod	C	[]
CCPackage	A_CC_CollectCallInd	C	[]
CCPackage	A_CC_oSCCPMethod	C	[]
CCPackage	A_CC_GDES	C	[]
CCPackage	A_CC_GDTD	C	[]
CCPackage	A_CC_NOCI_VC	C	[]
CCPackage	A_CC_NOCI_ECDI	C	[]
CCPackage	A_CC_NOCI_CCI	C	[]
CCPackage	A_CC_NOCI_SI	C	[]
CCPackage	A_CC_TMR	C	[]
CCPackage	A_CC_INFO_CFN	C	[]
CCPackage	A_CC_GAPPEDCALLCAUSE	C	[]
CCPackage	A_CC_WAIT_CONFIRM	C	[]
CCPackage	A_CC_WAIT_ANSWER	C	[]

```
CCPackage       A_CC_NUM_REL_RETRIES            C                       []
CCPackage       A_CC_WAIT_REL_RETRY             C                       []
CCPackage       A_CC_WAIT_RLC_FAIL              C                       []

# Cause code map

CCPackage       CC_EC_UnallocatedNumber         C                       STRING
CCPackage       CC_EC_NoRouteToTns              C                       STRING
CCPackage       CC_EC_NoRouteToDest             C                       STRING
CCPackage       CC_EC_SpecialInformationTone    C                       STRING
CCPackage       CC_EC_MisdialledTkPrefix        C                       STRING
CCPackage       CC_EC_ChUnacceptable            C                       STRING
CCPackage       CC_EC_CallAwardedDeliveredEstCh C                       STRING
CCPackage       CC_EC_Preemption                C                       STRING
CCPackage       CC_EC_PreemptionCctRes          C                       STRING
CCPackage       CC_EC_NormalClearing            C                       STRING
CCPackage       CC_EC_UserBusy                  C                       STRING
CCPackage       CC_EC_NoUserResponding          C                       STRING
CCPackage       CC_EC_NoAnswerAlertedUser       C                       STRING
CCPackage       CC_EC_SubAbscent                C                       STRING
CCPackage       CC_EC_CallRejected              C                       STRING
CCPackage       CC_EC_NumberChanged             C                       STRING
CCPackage       CC_EC_RedirectionToNewDest      C                       STRING
CCPackage       CC_EC_RoutingError              C                       STRING
CCPackage       CC_EC_NonSelectedUserClearing   C                       STRING
CCPackage       CC_EC_DestOutOfOrder            C                       STRING
CCPackage       CC_EC_InvalidNumberFormat       C                       STRING
CCPackage       CC_EC_FacilityRejected          C                       STRING
CCPackage       CC_EC_ResponseToStatusEnquiry   C                       STRING
CCPackage       CC_EC_NormalUnspecified         C                       STRING
CCPackage       CC_EC_NoCircuitAvailable        C                       STRING
CCPackage       CC_EC_NetworkOutOfOrder         C                       STRING
CCPackage       CC_EC_PermanentFrameModeOos     C                       STRING
CCPackage       CC_EC_PermanentFrameModeOperational C                   STRING
CCPackage       CC_EC_TemporaryFailure          C                       STRING
CCPackage       CC_EC_SwitchingEquipCongestion  C                       STRING
CCPackage       CC_EC_AccessInfoDiscarded       C                       STRING
CCPackage       CC_EC_ReqCircuitUnavail         C                       STRING
```

CCPackage	CC_EC_PrecedenceBlocked	C		STRING
CCPackage	CC_EC_ResourcesUnavailUnspec	C		STRING
CCPackage	CC_EC_QualityUnavail	C		STRING
CCPackage	CC_EC_ReqFacilityNotSubscr	C		STRING
CCPackage	CC_EC_OutgoingCallsBarredInCug	C		STRING
CCPackage	CC_EC_IncomingCallsBarredInCug	C		STRING
CCPackage	CC_EC_BearcapNotAuthorized	C		STRING
CCPackage	CC_EC_BaercapNotAvail	C		STRING
CCPackage	CC_EC_InconOutgoingAccAndSubClass	C		STRING
CCPackage	CC_EC_ServiceOrOptionNotAvail	C		STRING
CCPackage	CC_EC_BearcapNotImp	C		STRING
CCPackage	CC_EC_ChTypeNotImp	C		STRING
CCPackage	CC_EC_ReqFacilityNotImp	C		STRING
CCPackage	CC_EC_OnlyRestrictDigInfoBearer	C		STRING
CCPackage	CC_EC_ServiceOrOptionNotImpUnspec	C		STRING
CCPackage	CC_EC_InvalidCallReferenceValue	C		STRING
CCPackage	CC_EC_ChIdNotExist	C		STRING
CCPackage	CC_EC_SuspendExistButNotThisId	C		STRING
CCPackage	CC_EC_CallIdInUse	C		STRING
CCPackage	CC_EC_NoCallSuspended	C		STRING
CCPackage	CC_EC_CallIdHasBeenCleared	C		STRING
CCPackage	CC_EC_UserNotMemberOfCug	C		STRING
CCPackage	CC_EC_IncompatibleDest	C		STRING
CCPackage	CC_EC_NonExistentCug	C		STRING
CCPackage	CC_EC_InvalidTns	C		STRING
CCPackage	CC_EC_InvalidMsgUnspec	C		STRING
CCPackage	CC_EC_MandatoryElementMissing	C		STRING
CCPackage	CC_EC_MsgTypeNotImp		C	STRING
CCPackage	CC_EC_MsgTypeNotImpOrWrongState		C	STRING
CCPackage	CC_EC_ElemTypeNotImp		C	STRING
CCPackage	CC_EC_InvalidElemContents		C	STRING
CCPackage	CC_EC_MsgInWrongState		C	STRING
CCPackage	CC_EC_RecoveryOnTimerExpiry		C	STRING
CCPackage	CC_EC_ParamUnrecPassed		C	STRING
CCPackage	CC_HC_UnallocatedNumber		C	STRING
CCPackage	CC_HC_NoRouteToSpecifiedTransitNetwork		C	STRING
CCPackage	CC_HC_NoRouteToDestination		C	STRING
CCPackage	CC_HC_SendSpecialInformationTone		C	STRING

CCPackage	CC_HC_MisdialedTrunkPrefix	C	STRING
CCPackage	CC_HC_ChannelUnacceptable	C	STRING
CCPackage	CC_HC_CallAwardedEstablishedChannel	C	STRING
CCPackage	CC_HC_Preemption	C	STRING
CCPackage	CC_HC_PreemptionCircuitReservedForReuse	C	STRING
CCPackage	CC_HC_NormalCallClearing	C	STRING
CCPackage	CC_HC_UserBusy	C	STRING
CCPackage	CC_HC_NoUserresponding	C	STRING
CCPackage	CC_HC_NoAnswerFromAlertedUser	C	STRING
CCPackage	CC_HC_SubscriberAbsent	C	STRING
CCPackage	CC_HC_CallRejected	C	STRING
CCPackage	CC_HC_NumberChanged	C	STRING
CCPackage	CC_HC_RedirectionToNewDestination	C	STRING
CCPackage	CC_HC_ExchangeRouteError	C	STRING
CCPackage	CC_HC_NonSelectedUserClearing	C	STRING
CCPackage	CC_HC_DestinationOutOfOrder	C	STRING
CCPackage	CC_HC_InvalidNumberFormat	C	STRING
CCPackage	CC_HC_FacilityRejected	C	STRING
CCPackage	CC_HC_ResponceToStatusEnquiry	C	STRING
CCPackage	CC_HC_NormalUnspecified	C	STRING
CCPackage	CC_HC_NoCircuitChannelAvailable	C	STRING
CCPackage	CC_HC_NetworkOutOfOrder	C	STRING
CCPackage	CC_HC_PermanentFrameModeConnectionOutOfService	C	STRING
CCPackage	CC_HC_PermanentFrameModeConnectionOperational	C	STRING
CCPackage	CC_HC_TemporaryFailure	C	STRING
CCPackage	CC_HC_SwitchingEquipmentCongestion	C	STRING
CCPackage	CC_HC_AccessInformationDiscarded	C	STRING
CCPackage	CC_HC_RequestedCircuitChannelNotAvailable	C	STRING
CCPackage	CC_HC_PrecedenceCallBlocked	C	STRING
CCPackage	CC_HC_ResourceUnavailable	C	STRING
CCPackage	CC_HC_QualityOfServiceNotAvailable	C	STRING
CCPackage	CC_HC_RequestedFacilityNotSubscribed	C	STRING
CCPackage	CC_HC_OutgoingCallsBaredWithinCUG	C	STRING
CCPackage	CC_HC_IncomingCallsBaredWithinCUG	C	STRING
CCPackage	CC_HC_BearerCapabilityNotAuthorized	C	STRING
CCPackage	CC_HC_BearerCapabilityNotPresentlyAvailable	C	STRING
CCPackage	CC_HC_InconsistencyAcessInfoSubscriberClass	C	STRING
CCPackage	CC_HC_ServiceOrOptionUnavailable	C	STRING

```
CCPackage       CC_HC_BearerCapabilityNotImplemented                    C       STRING
CCPackage       CC_HC_ChannelTypeNotImplemented                         C       STRING
CCPackage       CC_HC_RequestedFacilityNotImplemented                   C       STRING
CCPackage       CC_HC_OnlyRestrictedDigitalBearerInfoCapability         C       STRING
CCPackage       CC_HC_ServiceOrOptionNotImplemented                     C       STRING
CCPackage       CC_HC_InvalidCallreferenceValue                         C       STRING
CCPackage       CC_HC_IdentifiedChannelDoesnotExist                     C       STRING
CCPackage       CC_HC_ASuspendedCallExistsThisCallIdDoesNot             C       STRING
CCPackage       CC_HC_CallIdentityInUse                                 C       STRING
CCPackage       CC_HC_NoCallSuspended                                   C       STRING
CCPackage       CC_HC_CallHavingTheRequestedCallIdHasBeenCleared        C       STRING
CCPackage       CC_HC_UserNotMemberOfCUG                                C       STRING
CCPackage       CC_HC_IncompatibleDestination                           C       STRING
CCPackage       CC_HC_NonExistantCUG                                    C       STRING
CCPackage       CC_HC_InvalidTransitNetworkSelection                    C       STRING
CCPackage       CC_HC_InvalidMessage                                    C       STRING
CCPackage       CC_HC_MandatoryInformationElementIsMissing              C       STRING
CCPackage       CC_HC_MessageTypeNonExistantOrNotImplemented            C       STRING
CCPackage       CC_HC_MessageTypeNotCompatibleWithStateOrNonExistantOrNotImplemented C
STRING
CCPackage       CC_HC_InformationElementParameterNonExistantOrNotImplemented C  STRING
CCPackage       CC_HC_InvalidInformationElementContents                 C       STRING
CCPackage       CC_HC_MessageNotCompatibleWithCallState                 C       STRING
CCPackage       CC_HC_RecoveryOnTimerExpiry                             C       STRING
CCPackage       CC_HC_ParameterNonExistantOrNotImplementedPassedOn      C       STRING
```

```
#
# Package Gapping
# Set the gapping percentage level for each side. A level of 0
# indicates no gapping
# A level of 100 indicates gap all calls (except priority calls -
# see treatment below)
#

Gapping         H323level       C       []
Gapping         EISUPlevel      C       []
```

```
#
# Priority treatment determines the treatment of priority calls
# during gapping.
# GapAlways indicates priority calls are treated as normal calls
# GapNever indicate priority calls are never to be gapped
# GapOn100PercentGapping indicates priority calls are only gapped
# when 100 percent gapping is applied.
#

Gapping              PriorityCallTreatment      C              STRING

#
# Configuration for trace package
# There are five trace trigger locations. Each location can hold one type of
# six trigger types  as follows:
#
#  EISUP CallingPartyNumber (E.164 address)
#  EISUP CalledPartyNumber (E.164 address)
#  H323 CallingPartyNumber (E.164 address)
#  H323 CalledPartyNumber (E.164 address)
#  H323 OriginatingIPAddress and H323 OriginatingIPMask (dotted notation)
#  H323 TerminatingIPAddress and TerminatingIPMask (dotted notation)
#

Trace    TraceOutputFilename                          C                    STRING
Trace    Trigger1.eisup.CallingPartyNumber            C                    []
Trace     Trigger1.eisup.CalledPartyNumber            C                    []
Trace     Trigger1.h323.CallingPartyNumber            C                    []
Trace     Trigger1.h323.CalledPartyNumber             C                    []
Trace     Trigger1.h323.OriginatingIPAddress          C                    []
Trace     Trigger1.h323.OriginatingIPMask             C                    []
Trace     Trigger1.h323.TerminatingIPAddress          C                    []
Trace     Trigger1.h323.TerminatingIPMask             C                    []

Trace     Trigger2.eisup.CallingPartyNumber           C                    []
Trace     Trigger2.eisup.CalledPartyNumber            C                    []
Trace     Trigger2.h323.CallingPartyNumber            C                    []
```

```
Trace    Trigger2.h323.CalledPartyNumber                C          STRING
Trace    Trigger2.h323.OriginatingIPAddress             C          STRING
Trace    Trigger2.h323.OriginatingIPMask                C          STRING
Trace    Trigger2.h323.TerminatingIPAddress             C          STRING
Trace    Trigger2.h323.TerminatingIPMask                C          STRING

Trace    Trigger3.eisup.CallingPartyNumber              C          []
Trace    Trigger3.eisup.CalledPartyNumber               C          []
Trace    Trigger3.h323.CallingPartyNumber               C          []
Trace    Trigger3.h323.CalledPartyNumber                C          []
Trace    Trigger3.h323.OriginatingIPAddress             C          []
Trace    Trigger3.h323.OriginatingIPMask                C          []
Trace    Trigger3.h323.TerminatingIPAddress             C          []
Trace    Trigger3.h323.TerminatingIPMask                C          []

Trace    Trigger4.eisup.CallingPartyNumber              C          []
Trace    Trigger4.eisup.CalledPartyNumber               C          []
Trace    Trigger4.h323.CallingPartyNumber               C          []
Trace    Trigger4.h323.CalledPartyNumber                C          []
Trace    Trigger4.h323.OriginatingIPAddress             C          []
Trace    Trigger4.h323.OriginatingIPMask                C          []
Trace    Trigger4.h323.TerminatingIPAddress             C          []
Trace    Trigger4.h323.TerminatingIPMask                C          []

Trace    Trigger5.eisup.CallingPartyNumber              C          []
Trace    Trigger5.eisup.CalledPartyNumber               C          []
Trace    Trigger5.h323.CallingPartyNumber               C          []
Trace    Trigger5.h323.CalledPartyNumber                C          []
Trace    Trigger5.h323.OriginatingIPAddress             C          STRING
Trace    Trigger5.h323.OriginatingIPMask                C          STRING
Trace    Trigger5.h323.TerminatingIPAddress             C          STRING
Trace    Trigger5.h323.TerminatingIPMask                C          STRING

#
# TraceTriggerSwitch(for CLI/SNMP application)
# This gates the output of the trigger data for each location
#
Trace    TriggerGate1                                   C          STRING
```

```
Trace     TriggerGate2                                  C                 STRING
Trace     TriggerGate3                                  C                 STRING
Trace     TriggerGate4                                  C                 STRING
Trace     TriggerGate5                                  C                 STRING


##
## Package EISUP
##

#
#The period for CISCO's RUDP timer manager in milli seconds
#
EISUP     RUDP_TIMER_CHECK_PERIOD_MSEC                  C                 []

#The time to wait before failing over to another vsc.
EISUP     WAIT_TIME_BEFORE_FAIL_OVER_MILLI_SEC          C                 []

##
## Package Application
##

Application     DefaultCallProcessingStatus             C                 STRING
Application     WaitBeforeCallReleaseTimer              C                 []
Application     RestartPendingTimer                     C                 []
Application     HaltPendingTimer                        C                 []
Application     RebootPendingTimer                      C                 []


#
#Information required by the rtrv-ne-health MML command
#

Application     Hardware                                C                 STRING
Application      Location                               C                 STRING

##
## Package H323
```

```
##
H323      maxTimers                C      []
H323      tickPeriod               C      []
H323      defaultSDPt              C      STRING
H323      overrideConfig           S      [0-1]
H323      defaultRadLog            S      [0-4]

##
## Statistics
##
Statistics      StatLogFileName    D      STRING
Statistics      StatOutputTime     D      [0-65535]

#
# Test entries for provisioning tests.
#

RAS       NumberNoRangeConst       C      []
RAS       NumberRangeDynamic2_4    D      [2-4]

H245      NumberNoRangeConst       C      []
H245      NumberRangeStatic1_4     S      [1-4]

Q931      QuotedDynamic            D      STRING
Q931      NonQuotedConst           C      STRING
Q931      ConnectTimeoutConst      C      STRING
```

Example of an HSI Configuration File

This appendix presents an example of an HSI configuration file.

Note The configuration file does not contain a complete list of all configurable items.

```
#verified
#
# $Id: GWmain.base.conf,v 1.6 2002/09/09 15:27:10 syousaf Exp $
#

# This is the base configuration file that is concatenated to
# a file derived from questions at install time, to generate the GWmain.conf
# file, which is used by the Application GWmain.


##############################################################################
##
# LOGGING PACKAGE
#
# The Logging package determines the logging level for all defined packages.
# This is a bit mask which controls the 16 debug levels
#
Package = Logging
#
OTLogging       = "ON"                  #Choice {ON, OFF}. Default: "OFF"

Application          = 0x0000
#CallControl         = 0xFF00
CallControl          = 0x0000
```

```
Connection          = 0x0000
#Connection         = 0xFF00
DataManager         = 0x0000
#Eisup              = 0xFF00
Eisup               = 0x0000
FaultManager        = 0x0000
Gapping             = 0x0000
H323                = 0x0000
Infrastructure      = 0x0000
Overload            = 0x0000
ProcessManager      = 0x0000
Provisioning        = 0x0000
Signal              = 0x0000
Snmp                = 0x0000
SnmpSubagent        = 0x0000
Statistics          = 0x0000
Trace               = 0x0000
UserInterface       = 0x0000


################################################################################
##
# H323 Service Package
#
# Not modifiable at runtime (Static Provisionable Data)
#
Package = H323
#
maxTimers = 20
tickPeriod = 1000
defaultSDPt = "v=0\r\no=\r\ns=\r\nt=0 0\r\nc=IN IP4 0.0.0.0\r\nm=audio 0 RTP/AVP 0 8\r\n"
overrideConfig = 0    # Set to 1 to use the override_config.val file instead of RVConfig
defaultRadLog = 0     # Set to 3(or any rad log level) to start radvision logging at startup.

################################################################################
##
# H323 RADVision SYSTEM Package
#
```

```
# Not modifiable at runtime (Static Provisionable Data)
#
Package = H323_SYS
#
maxCalls = 2500
maxChannels = 2



##########################################################################################
##
# Q931 RADVIsion PACKAGE
#
Package = Q931
#
responseTimeOut = 60
connectTimeOut = 180
callSignalingPort = 1720
maxCalls = 2500
#notEstablishControl =
overlappedSending =
earlyH245 =
h245Tunneling =



##########################################################################################
###
# H323 RADVision RAS Package
#
# Modifiable at runtime (Dynamic Provisionable Data) except for manualRAS
#
Package = RAS
#
responseTimeOut = 30
#manualRAS =
maxFail = 3
#allowCallsWhenNonReg =
```

```
#manualRegistration =
endpointVendor.productID = "GoldWing"
endpointVendor.t35CountryCode = 11
endpointVendor.t35Extension = 11
endpointVendor.manufacturerCode = 9
timeToLive = 600
rasPort = 0
#compare15bitRasCrv =
maxRetries = 3
maxMulticastTTL = 3
preGrantedArqUse = direct



################################################################################################
####
# H245 RADVision PACKAGE
#
# Dynamically Provisionable except for manualOperation(s)
Package = H245
#
channelsTimeout = 30
#roundTripTimeout = 5
#requestCloseTimeout = 5
#requestModeTimeout = 5
#mediaLoopTimeout = 5
## MasterSlave Determination
masterSlave.terminalType = 60
masterSlave.manualOperation =
masterSlave.manualResponse =
masterSlave.timeout = 30
caps.manualOperation =
caps.timeout = 30
caps.maxAudioDelay = 60
caps.table[1].entryNo = 7111
caps.table[1].audio.g711Ulaw64k = 20
caps.table[2].entryNo = 7110
caps.table[2].audio.g711Alaw64k = 20
```

```
caps.table[3].entryNo = 728
caps.table[3].audio.g728 = 20
chan[1].name = g711Alaw64k
chan[1].audio.g711Alaw64k = 20
chan[2].name = g711Ulaw64k
chan[2].audio.g711Ulaw64k = 20
chan[3].name = g728
chan[3].audio.g728 = 20
modes[1].name = g711Alaw64k
modes[1].audio.g711Alaw64k =
modes[2].name = g711Ulaw64k
modes[2].audio.g711Ulaw64k =
#modes[3].name = g728
#modes[3].audio.g728 =



################################################################################
##
# CALL CONTROL PACKAGE
#
##
Package = CCPackage
#
Hash = A
Pound = A
Star = B
StopDigit = "#"

A_CC_ChargeInd =  # BCI
A_CC_tEndToEndMethod =
A_CC_tLineUser =
A_CC_tLineStatus =
A_CC_MLC_Action =
A_CC_tSCCPMethod =
A_CC_Interworking =
A_CC_tEndToEndInfAvail =
A_CC_tIsdnAllTheWay =
```

```
A_CC_tEchoCancIr =
A_CC_tLineAccess =
A_CC_BNumDataNOA =              # CalledPN
A_CC_BNumDataNPI =
A_CC_BNumDataINN =
A_CC_ANumDataNOA =              # CallingPN
A_CC_Clir =
A_CC_ANumDataSI =
A_CC_ANumDataNPI =
A_CC_A_Cli =
A_CC_oLinecall =  # CallingPC
A_CC_Location =   # CauseInd
A_CC_CodeStandard =
A_CC_ProgressRestrict =         # Event Info
A_CC_oIsdnPref =  # FCI
A_CC_oIsdnAllTheWay =
A_CC_oEndToEndInfAvail =
A_CC_oNatInd =
A_CC_oLSPP =
A_CC_oNBit =
A_CC_oPORC =
A_CC_oPBit =
A_CC_oEndToEndMethod =
A_CC_CollectCallInd =
A_CC_oSCCPMethod =
A_CC_GDES =       # GenericDigits
A_CC_GDTD =
A_CC_NOCI_VC =    # NatureOfConnection
A_CC_NOCI_ECDI =
A_CC_NOCI_CCI =
A_CC_NOCI_SI =
A_CC_TMR =        # TransmissionMediumRequired
A_CC_INFO_CFN =   # confusion code on INFO receipt
A_CC_GAPPEDCALLCAUSE =  60 # congestion cause for releasing on gapping
A_CC_WAIT_CONFIRM = 30 #20..30 seconds (default is 30), from q764
A_CC_WAIT_ANSWER = 180 #90..180 seconds (default is 180), from q118, refd in

# ----- Cause Codes ----------------------------------------------------------------------------------------
```

```
# CC: Call Control, EC: Eisup Cause, HC: H323 Cause

# For the H323 cause code values see ITU-T: Q.850

# The mappings below are considered constant and not provisionable.
# They can be made provisionable by moving them from the CCPackage
# to the SYS_CONFIG_STATIC package.


# The following is the Eisup to H323 cause code map.
# When the Eisup cause on the left is received from Eisup,
# the H323 cause on right is sent to H323.
# Note: the reverse is not true, this is a one way mapping.
#       The H323 to Eisup cause map is defined further down.

CC_EC_UnallocatedNumber             = CC_HC_UnallocatedNumber
CC_EC_NoRouteToTns                  = CC_HC_NoRouteToSpecifiedTransitNetwork
CC_EC_NoRouteToDest                 = CC_HC_NoRouteToDestination
CC_EC_SpecialInformationTone        = CC_HC_SendSpecialInformationTone
CC_EC_MisdialledTkPrefix            = CC_HC_MisdialedTrunkPrefix
CC_EC_ChUnacceptable                = CC_HC_ChannelUnacceptable
CC_EC_CallAwardedDeliveredEstCh     = CC_HC_CallAwardedEstablishedChannel
CC_EC_Preemption                    = CC_HC_Preemption
CC_EC_PreemptionCctRes              = CC_HC_PreemptionCircuitReservedForReuse
CC_EC_NormalClearing                = CC_HC_NormalCallClearing
CC_EC_UserBusy                      = CC_HC_UserBusy
CC_EC_NoUserResponding              = CC_HC_NoUserresponding
CC_EC_NoAnswerAlertedUser           = CC_HC_NoAnswerFromAlertedUser
CC_EC_SubAbscent                    = CC_HC_SubscriberAbsent
CC_EC_CallRejected                  = CC_HC_CallRejected
CC_EC_NumberChanged                 = CC_HC_NumberChanged
CC_EC_RedirectionToNewDest          = CC_HC_RedirectionToNewDestination
CC_EC_RoutingError                  = CC_HC_ExchangeRouteError
CC_EC_NonSelectedUserClearing       = CC_HC_NonSelectedUserClearing
CC_EC_DestOutOfOrder                = CC_HC_DestinationOutOfOrder
CC_EC_InvalidNumberFormat           = CC_HC_InvalidNumberFormat
CC_EC_FacilityRejected              = CC_HC_FacilityRejected
CC_EC_ResponseToStatusEnquiry       = CC_HC_ResponceToStatusEnquiry
```

```
CC_EC_NormalUnspecified              = CC_HC_NormalUnspecified
CC_EC_NoCircuitAvailable             = CC_HC_NoCircuitChannelAvailable
CC_EC_NetworkOutOfOrder              = CC_HC_NetworkOutOfOrder
CC_EC_PermanentFrameModeOos          = CC_HC_PermanentFrameModeConnectionOutOfService
CC_EC_PermanentFrameModeOperational = CC_HC_PermanentFrameModeConnectionOperational
CC_EC_TemporaryFailure                = CC_HC_TemporaryFailure
CC_EC_SwitchingEquipCongestion       = CC_HC_SwitchingEquipmentCongestion
CC_EC_AccessInfoDiscarded             = CC_HC_AccessInformationDiscarded
CC_EC_ReqCircuitUnavail              = CC_HC_RequestedCircuitChannelNotAvailable
CC_EC_PrecedenceBlocked              = CC_HC_PrecedenceCallBlocked
CC_EC_ResourcesUnavailUnspec         = CC_HC_ResourceUnavailable
CC_EC_QualityUnavail                 = CC_HC_QualityOfServiceNotAvailable
CC_EC_ReqFacilityNotSubscr           = CC_HC_RequestedFacilityNotSubscribed
CC_EC_OutgoingCallsBarredInCug       = CC_HC_OutgoingCallsBaredWithinCUG
CC_EC_IncomingCallsBarredInCug       = CC_HC_IncomingCallsBaredWithinCUG
CC_EC_BearcapNotAuthorized           = CC_HC_BearerCapabilityNotAuthorized
CC_EC_BaercapNotAvail                = CC_HC_BearerCapabilityNotPresentlyAvailable
CC_EC_InconOutgoingAccAndSubClass    = CC_HC_InconsistencyAcessInfoSubscriberClass
CC_EC_ServiceOrOptionNotAvail        = CC_HC_ServiceOrOptionUnavailable
CC_EC_BearcapNotImp                   = CC_HC_BearerCapabilityNotImplemented
CC_EC_ChTypeNotImp                   = CC_HC_ChannelTypeNotImplemented
CC_EC_ReqFacilityNotImp              = CC_HC_RequestedFacilityNotImplemented
CC_EC_OnlyRestrictDigInfoBearer      = CC_HC_OnlyRestrictedDigitalBearerInfoCapability
CC_EC_ServiceOrOptionNotImpUnspec    = CC_HC_ServiceOrOptionNotImplemented
CC_EC_InvalidCallReferenceValue      = CC_HC_InvalidCallreferenceValue
CC_EC_ChIdNotExist                    = CC_HC_IdentifiedChannelDoesnotExist
CC_EC_SuspendExistButNotThisId       = CC_HC_ASuspendedCallExistsThisCallIdDoesNot
CC_EC_CallIdInUse                     = CC_HC_CallIdentityInUse
CC_EC_NoCallSuspended                = CC_HC_NoCallSuspended
CC_EC_CallIdHasBeenCleared           = CC_HC_CallHavingTheRequestedCallIdHasBeenCleared
CC_EC_UserNotMemberOfCug              = CC_HC_UserNotMemberOfCUG
CC_EC_IncompatibleDest               = CC_HC_IncompatibleDestination
CC_EC_NonExistentCug                 = CC_HC_NonExistantCUG
CC_EC_InvalidTns                      = CC_HC_InvalidTransitNetworkSelection
CC_EC_InvalidMsgUnspec               = CC_HC_InvalidMessage
CC_EC_MandatoryElementMissing        = CC_HC_MandatoryInformationElementIsMissing
CC_EC_MsgTypeNotImp                   = CC_HC_MessageTypeNonExistantOrNotImplemented
```

```
CC_EC_MsgTypeNotImpOrWrongState =
CC_HC_MessageTypeNotCompatibleWithStateOrNonExistantOrNotImplemented
CC_EC_ElemTypeNotImp = CC_HC_InformationElementParameterNonExistantOrNotImplemented
CC_EC_InvalidElemContents          = CC_HC_InvalidInformationElementContents
CC_EC_MsgInWrongState              = CC_HC_MessageNotCompatibleWithCallState
CC_EC_RecoveryOnTimerExpiry        = CC_HC_RecoveryOnTimerExpiry
CC_EC_ParamUnrecPassed             = CC_HC_ParameterNonExistantOrNotImplementedPassedOn


# When the H323 cause on the left is received from H323,
# the Eisup cause on the right is sent to Eisup.
# Note: the reverse is not true, this is a one way mapping.
#       The Eisup to H323 cause map is defined above.

CC_HC_UnallocatedNumber                 = CC_EC_UnallocatedNumber
CC_HC_NoRouteToSpecifiedTransitNetwork  = CC_EC_NoRouteToTns
CC_HC_NoRouteToDestination              = CC_EC_NoRouteToDest
CC_HC_SendSpecialInformationTone        = CC_EC_SpecialInformationTone
CC_HC_MisdialedTrunkPrefix              = CC_EC_MisdialledTkPrefix
CC_HC_ChannelUnacceptable               = CC_EC_ChUnacceptable
CC_HC_CallAwardedEstablishedChannel     = CC_EC_CallAwardedDeliveredEstCh
CC_HC_Preemption                        = CC_EC_Preemption
CC_HC_PreemptionCircuitReservedForReuse = CC_EC_PreemptionCctRes
CC_HC_NormalCallClearing                = CC_EC_NormalClearing
CC_HC_UserBusy                          = CC_EC_UserBusy
CC_HC_NoUserresponding                  = CC_EC_NoUserResponding
CC_HC_NoAnswerFromAlertedUser           = CC_EC_NoAnswerAlertedUser
CC_HC_SubscriberAbsent                  = CC_EC_SubAbscent
CC_HC_CallRejected                      = CC_EC_CallRejected
CC_HC_NumberChanged                     = CC_EC_NumberChanged
CC_HC_RedirectionToNewDestination       = CC_EC_RedirectionToNewDest
CC_HC_ExchangeRouteError                = CC_EC_RoutingError
CC_HC_NonSelectedUserClearing           = CC_EC_NonSelectedUserClearing
CC_HC_DestinationOutOfOrder             = CC_EC_DestOutOfOrder
CC_HC_InvalidNumberFormat               = CC_EC_InvalidNumberFormat
CC_HC_FacilityRejected                  = CC_EC_FacilityRejected
CC_HC_ResponceToStatusEnquiry           = CC_EC_ResponseToStatusEnquiry
CC_HC_NormalUnspecified                 = CC_EC_NormalUnspecified
```

CC_HC_NoCircuitChannelAvailable = CC_EC_NoCircuitAvailable
CC_HC_NetworkOutOfOrder = CC_EC_NetworkOutOfOrder
CC_HC_PermanentFrameModeConnectionOutOfService = CC_EC_PermanentFrameModeOos
CC_HC_PermanentFrameModeConnectionOperational = CC_EC_PermanentFrameModeOperational
CC_HC_TemporaryFailure = CC_EC_TemporaryFailure
CC_HC_SwitchingEquipmentCongestion = CC_EC_SwitchingEquipCongestion
CC_HC_AccessInformationDiscarded = CC_EC_AccessInfoDiscarded
CC_HC_RequestedCircuitChannelNotAvailable = CC_EC_ReqCircuitUnavail
CC_HC_PrecedenceCallBlocked = CC_EC_PrecedenceBlocked
CC_HC_ResourceUnavailable = CC_EC_ResourcesUnavailUnspec
CC_HC_QualityOfServiceNotAvailable = CC_EC_QualityUnavail
CC_HC_RequestedFacilityNotSubscribed = CC_EC_ReqFacilityNotSubscr
CC_HC_OutgoingCallsBaredWithinCUG = CC_EC_OutgoingCallsBarredInCug
CC_HC_IncomingCallsBaredWithinCUG = CC_EC_IncomingCallsBarredInCug
CC_HC_BearerCapabilityNotAuthorized = CC_EC_BearcapNotAuthorized
CC_HC_BearerCapabilityNotPresentlyAvailable = CC_EC_BaercapNotAvail
CC_HC_InconsistencyAcessInfoSubscriberClass = CC_EC_InconOutgoingAccAndSubClass
CC_HC_ServiceOrOptionUnavailable = CC_EC_ServiceOrOptionNotAvail
CC_HC_BearerCapabilityNotImplemented = CC_EC_BearcapNotImp
CC_HC_ChannelTypeNotImplemented = CC_EC_ChTypeNotImp
CC_HC_RequestedFacilityNotImplemented = CC_EC_ReqFacilityNotImp
CC_HC_OnlyRestrictedDigitalBearerInfoCapability = CC_EC_OnlyRestrictDigInfoBearer
CC_HC_ServiceOrOptionNotImplemented = CC_EC_ServiceOrOptionNotImpUnspec
CC_HC_InvalidCallreferenceValue = CC_EC_InvalidCallReferenceValue
CC_HC_IdentifiedChannelDoesnotExist = CC_EC_ChIdNotExist
CC_HC_ASuspendedCallExistsThisCallIdDoesNot = CC_EC_SuspendExistButNotThisId
CC_HC_CallIdentityInUse = CC_EC_CallIdInUse
CC_HC_NoCallSuspended = CC_EC_NoCallSuspended
CC_HC_CallHavingTheRequestedCallIdHasBeenCleared = CC_EC_CallIdHasBeenCleared
CC_HC_UserNotMemberOfCUG = CC_EC_UserNotMemberOfCug
CC_HC_IncompatibleDestination = CC_EC_IncompatibleDest
CC_HC_NonExistantCUG = CC_EC_NonExistentCug
CC_HC_InvalidTransitNetworkSelection = CC_EC_InvalidTns
CC_HC_InvalidMessage = CC_EC_InvalidMsgUnspec
CC_HC_MandatoryInformationElementIsMissing = CC_EC_MandatoryElementMissing
CC_HC_MessageTypeNonExistantOrNotImplemented = CC_EC_MsgTypeNotImp
CC_HC_MessageTypeNotCompatibleWithStateOrNonExistantOrNotImplemented = CC_EC_MsgTypeNotImpOrWrongState

```
CC_HC_InformationElementParameterNonExistantOrNotImplemented = CC_EC_ElemTypeNotImp
CC_HC_InvalidInformationElementContents              = CC_EC_InvalidElemContents
CC_HC_MessageNotCompatibleWithCallState              = CC_EC_MsgInWrongState
CC_HC_RecoveryOnTimerExpiry                            = CC_EC_RecoveryOnTimerExpiry
CC_HC_ParameterNonExistantOrNotImplementedPassedOn = CC_EC_ParamUnrecPassed


# ------------------------------------------------------------------------------------------------------------


##############################################################################
##
# FAULTMANAGEMENT PACKAGE
#
Package = FaultManagement
#
FMRaiseRecoveryAction  =   "ON"
FMClearRecoveryAction  =   "ON"


##############################################################################
##
# GAPPING PACKAGE
#
# Set the gapping percentage level for each side. A level of 0 indicates no gapping
# A level of 100 indicates gap all calls (except priority calls - see treatment below)
#
Package = Gapping
#
H323level = 0
EISUPlevel = 0

#
# Priority treatment determines the treatment of priority calls during gapping.
# GapAlways indicates priority calls are treated as normal calls
# GapNever indicate priority calls are never to be gapped
```

```
# GapOn100PercentGapping indicates priority calls are only gapped when 100 percent
# gapping is applied.
#

#PriorityCallTreatment = GapOn100PercentGapping
#PriorityCallTreatment = GapNever
PriorityCallTreatment = GapAlways



##############################################################################################
##
# TRACE PACKAGE
#
# There are five trace trigger locations. Each location can hold one type of
# six trigger types  as follows:
#
#  EISUP CallingPartyNumber (E.164 address)
#  EISUP CalledPartyNumber (E.164 address)
#  H323 CallingPartyNumber (E.164 address)
#  H323 CalledPartyNumber (E.164 address)
#  H323 OriginatingIPAddress and H323 OriginatingIPMask (dotted notation)
#  H323 TerminatingIPAddress and TerminatingIPMask (dotted notation)
#
Package = Trace
#
TraceOutputFilename = GWtrace.txt

Trigger1.eisup.CallingPartyNumber=1800
#Trigger1.eisup.CalledPartyNumber=1900
#Trigger1.h323.CallingPartyNumber=0299
#Trigger1.h323.CalledPartyNumber=0388
#Trigger1.h323.OriginatingIPAddress=203.188.2.3
#Trigger1.h323.OriginatingIPMask=255.255.0.0
#Trigger1.h323.TerminatingIPAddress=203.155.7.9
#Trigger1.h323.TerminatingIPMask=255.255.0.0

#Trigger2.eisup.CallingPartyNumber=1800
```

```
Trigger2.eisup.CalledPartyNumber=1900
#Trigger2.h323.CallingPartyNumber=0299
#Trigger2.h323.CalledPartyNumber=0388
#Trigger2.h323.OriginatingIPAddress=203.188.2.3
#Trigger2.h323.OriginatingIPMask=255.255.0.0
#Trigger2.h323.TerminatingIPAddress=203.15y5.7.9
#Trigger2.h323.TerminatingIPMask=255.255.0.0

#Trigger3.eisup.CallingPartyNumber=1800
#Trigger3.eisup.CalledPartyNumber=1900
#Trigger3.h323.CallingPartyNumber=0299
#Trigger3.h323.CalledPartyNumber=0388
Trigger3.h323.OriginatingIPAddress=203.188.2.3
Trigger3.h323.OriginatingIPMask=255.255.0.0
#Trigger3.h323.TerminatingIPAddress=203.155.7.9
#Trigger3.h323.TerminatingIPMask=255.255.0.0

#Trigger4.eisup.CallingPartyNumber=1800
#Trigger4.eisup.CalledPartyNumber=1900
#Trigger4.h323.CallingPartyNumber=0299
Trigger4.h323.CalledPartyNumber=0388
#Trigger4.h323.OriginatingIPAddress=203.188.2.3
#Trigger4.h323.OriginatingIPMask=255.255.0.0
#Trigger4.h323.TerminatingIPAddress=203.155.7.9
#Trigger4.h323.TerminatingIPMask=255.255.0.0

#Trigger5.eisup.CallingPartyNumber=1800
#Trigger5.eisup.CalledPartyNumber=1900
#Trigger5.h323.CallingPartyNumber=0299
#Trigger5.h323.CalledPartyNumber=0388
#Trigger5.h323.OriginatingIPAddress=203.188.2.3
#Trigger5.h323.OriginatingIPMask=255.255.0.0
Trigger5.h323.TerminatingIPAddress=203.155.7.9
Trigger5.h323.TerminatingIPMask=255.255.0.0

#
# TraceTriggerSwitch(for CLI/SNMP application)
# This gates the output of the trigger data for each location
```

```
#
TriggerGate1=ON
TriggerGate2=ON
TriggerGate3=ON
TriggerGate4=ON
TriggerGate5=ON



##############################################################################################
##
# EISUP PACKAGE
#
#
Package = EISUP
#
#The period for CISCO's RUDP timer manage, in milli seconds
#RUDP_TIMER_CHECK_PERIOD_MSEC=20

#The time to wait before failing over to another VSC.
WAIT_TIME_BEFORE_FAIL_OVER_MILLI_SEC=1000



##############################################################################################
##
# APPLICATION PACKAGE
#
#
Package = Application
#
DefaultCallProcessingStatus = "UP"  #Choice {"UP", "DOWN"}

WaitBeforeCallReleaseTimer  = 20    #Default is 60
RestartPendingTimer          = 20    #Default is 60
HaltPendingTimer             = 20    #Default is 60
RebootPendingTimer           = 20    #Default is 60
```

```
################################################################################
##
# DYNAMIC SYSTEM DATA
#
#
Package = SYS_CONFIG_DYNAMIC


#
# Alternate Gatekeeper
ALTERNATEGATEKEEPERIP    = "" #Leave blank if you don't want to provision an alternate
gatekeeper, otherwise insert IP address e.g. 10.70.54.55
ALTERNATEGATEKEEPERPORT  = 1719
ALTERNATEGATEKEEPERID    = "OuterLondonAlt"


# Logging
#
LOGDIRECTORY                = "var/log/"      #Default: "var/log/"
LOGFILENAMEPREFIX           = "platform"      #Default: "platform.log"
LOGPRIO                     = "TRACE"         #Choice {DEBUG, TRACE, INFO, WARN, ERR,
CRIT}. Default: "WARN"
LOGFILEROTATESIZE           = 10240           #Default: 10240 bytes (10Mb)
LOGFILEROTATEINTERVAL = 1440           #Default: 1440 min (24hrs)



# Overload
#
DISKUSAGELIMIT                  = 98         #Default: 95% Disk Usage
OVLDSAMPLERATE                  = 3000       #Default: 3000 msec polling rate

OVLDLEVEL1PERCENT               = 20         #Default: 0
OVLDLEVEL1FILTER                = "NORMAL"   #Choice {"NORMAL", "ALL"}. Default:
"NORMAL"
OVLDLEVEL1THRESHUPPERCPU        = 65         #Default: 100
OVLDLEVEL1THRESHLOWERCPU        = 60         #Default: 100
OVLDLEVEL1THRESHUPPERCALLS      = 1900       #Default: 1000
OVLDLEVEL1THRESHLOWERCALLS   = 1800       #Default: 1000

OVLDLEVEL2PERCENT               = 75         #Default: 0
```

```
OVLDLEVEL2FILTER                    = "NORMAL"    #Choice {"NORMAL", "ALL"}. Default:
"NORMAL"
OVLDLEVEL2THRESHUPPERCPU        = 80        #Default: 100
OVLDLEVEL2THRESHLOWERCPU        = 70        #Default: 100
OVLDLEVEL2THRESHUPPERCALLS      = 2200      #Default: 1000
OVLDLEVEL2THRESHLOWERCALLS      = 2000      #Default: 1000

OVLDLEVEL3PERCENT                   = 90        #Default: 0
OVLDLEVEL3FILTER                    = "NORMAL"    #Choice {"NORMAL", "ALL"}.
Default: "NORMAL"
OVLDLEVEL3THRESHUPPERCPU        = 95        #Default: 100
OVLDLEVEL3THRESHLOWERCPU        = 85        #Default: 100
OVLDLEVEL3THRESHUPPERCALLS      = 2400      #Default: 1000
OVLDLEVEL3THRESHLOWERCALLS      = 2300      #Default: 1000



################################################################################
##
#
Package = SYS_CONFIG_STATIC

#
# Call Control
# For the H323 cause code values see ITU-T: Q.850
# The default cause codes, used when there is no map entry for a received cause

CC_EC_DEFAULT                           = CC_EC_NormalUnspecified
CC_HC_DEFAULT                           = CC_HC_NormalUnspecified

#
# Unassigned Eisup cause codes
#

CC_EC_AccessBarred                      = CC_HC_DEFAULT
CC_EC_Acknowledgement                   = CC_HC_DEFAULT
CC_EC_AddressIncomplete                 = CC_HC_DEFAULT
CC_EC_AnonymousCallRejection            = CC_HC_DEFAULT
```

```
CC_EC_BlacklistBNumberMatched          = CC_HC_DEFAULT
CC_EC_BlacklistCliLengthInvalid        = CC_HC_DEFAULT
CC_EC_BlacklistCliMatched              = CC_HC_DEFAULT
CC_EC_BlacklistCpcRestricted           = CC_HC_DEFAULT
CC_EC_BlacklistNoCli                   = CC_HC_DEFAULT
CC_EC_BlacklistNoaRestricted           = CC_HC_DEFAULT
CC_EC_Busy                             = CC_HC_DEFAULT
CC_EC_CallRejectCallGapping            = CC_HC_DEFAULT
CC_EC_CallTerminated                   = CC_HC_DEFAULT
CC_EC_CallTypeIncompatible             = CC_HC_DEFAULT
CC_EC_CallingDroppedWhileOnHold        = CC_HC_DEFAULT
CC_EC_CallingPartyOffHold              = CC_HC_DEFAULT
CC_EC_ChannelOutOfService              = CC_HC_DEFAULT
CC_EC_Congestion                       = CC_HC_DEFAULT
CC_EC_CotFailure                       = CC_HC_DEFAULT
CC_EC_CugAccessBarred                  = CC_HC_DEFAULT
CC_EC_DteControlledNotReady            = CC_HC_DEFAULT
CC_EC_DteUncontrolledNotReady          = CC_HC_DEFAULT
CC_EC_ExcessiveDigCallProceeding       = CC_HC_DEFAULT
CC_EC_FacilityNotRegistered            = CC_HC_DEFAULT
CC_EC_FlowControlledCongestion         = CC_HC_DEFAULT
CC_EC_GroupRestrictions                = CC_HC_DEFAULT
CC_EC_IncomingCallsBarred              = CC_HC_DEFAULT
CC_EC_InterceptedSubscriber            = CC_HC_DEFAULT
CC_EC_InterworkUnspec                  = CC_HC_DEFAULT
CC_EC_InvalidCallRef                   = CC_HC_DEFAULT
CC_EC_MesgWithUnrecElemDiscarded       = CC_HC_DEFAULT
CC_EC_MessageNotUnderstood             = CC_HC_DEFAULT
CC_EC_MisroutedCallPortedNumber        = CC_HC_DEFAULT
CC_EC_NetworkAddressExtensionError     = CC_HC_DEFAULT
CC_EC_NetworkTermination               = CC_HC_DEFAULT
CC_EC_NewDestination                   = CC_HC_DEFAULT
CC_EC_NumberUnobtainable               = CC_HC_DEFAULT
CC_EC_OperatorPriorityAccess           = CC_HC_DEFAULT
CC_EC_OutOfCatchmentArea               = CC_HC_DEFAULT
CC_EC_OutgoingCallsBarred              = CC_HC_DEFAULT
CC_EC_PermanentIcb                     = CC_HC_DEFAULT
CC_EC_PortedNumber                     = CC_HC_DEFAULT
```

```
CC_EC_PreemptionCctUnavailable          = CC_HC_DEFAULT
CC_EC_Prefix0DialledInError             = CC_HC_DEFAULT
CC_EC_Prefix1DialledInError             = CC_HC_DEFAULT
CC_EC_Prefix1NotDialled                 = CC_HC_DEFAULT
CC_EC_PriorityForcedRelease             = CC_HC_DEFAULT
CC_EC_Proprietary                       = CC_HC_DEFAULT
CC_EC_ProtErrThresholdExceeded          = CC_HC_DEFAULT
CC_EC_ProtocolErrorUnspec               = CC_HC_DEFAULT
CC_EC_Reject                            = CC_HC_DEFAULT
CC_EC_RejectedDivertedCall              = CC_HC_DEFAULT
CC_EC_RemoteProcError                   = CC_HC_DEFAULT
CC_EC_RepeatAttempt                     = CC_HC_DEFAULT
CC_EC_RouteOutOfService                 = CC_HC_DEFAULT
CC_EC_SelectiveCallBarring              = CC_HC_DEFAULT
CC_EC_ServiceIncompatible               = CC_HC_DEFAULT
CC_EC_ServiceTemporarilyUnavailable     = CC_HC_DEFAULT
CC_EC_ServiceUnavailable                = CC_HC_DEFAULT
CC_EC_SignalNotUnderstood               = CC_HC_DEFAULT
CC_EC_SignalNotValid                    = CC_HC_DEFAULT
CC_EC_SignallingSystemIncompatible      = CC_HC_DEFAULT
CC_EC_SubControlledIcb                  = CC_HC_DEFAULT
CC_EC_SubNotFoundDle                    = CC_HC_DEFAULT
CC_EC_SubscriberCallTerminate           = CC_HC_DEFAULT
CC_EC_SubscriberIncompatible            = CC_HC_DEFAULT
CC_EC_SubscriberMoved                   = CC_HC_DEFAULT
CC_EC_SubscriberOutOfService            = CC_HC_DEFAULT
CC_EC_TemporaryOos                      = CC_HC_DEFAULT
CC_EC_TerminalCongestion                = CC_HC_DEFAULT
CC_EC_Transferred                       = CC_HC_DEFAULT
CC_EC_TranslationOos                    = CC_HC_DEFAULT
CC_EC_UnallocatedDestNumber             = CC_HC_DEFAULT
CC_EC_UndefinedBg                       = CC_HC_DEFAULT
CC_EC_Unknown                           = CC_HC_DEFAULT
CC_EC_UnrecElemPassedOn                 = CC_HC_DEFAULT
CC_EC_VacentCode                        = CC_HC_DEFAULT
CC_EC_WhitelistCliNotMatched            = CC_HC_DEFAULT

#
```

```
# T38 Fax default configuration
#
T38MaxVal = "MaxBit 0x90, FxMaxBuf 0xc8, FxMaxData 0x48"
T38Options = "FxFillBit 0, FxTransMMR 0, FxTransJBIG 0, FxRate Trans, FxUdpEC Red"

#
#
# EISUP Settings for GoldWing to look at EISUP Test Tool
# Point GWmain to look at the test tool HOST_PORT instead of the VSC's
#VSCA_IPADDR1=samson
#VSCA_PORT_NUMBER1=18613
#VSCB_IPADDR1=stonehenge
#VSCB_PORT_NUMBER1=18613
```

E-ISUP Name-to-Cause Value Lookup

This appendix lists the Cisco E-ISUP protocol names and their associated cause values.

Name	Value
CC_EC_AccessBarred	54
CC_EC_AccessInfoDiscarded	1
CC_EC_Acknowledgement	55
CC_EC_AddressIncomplete	56
CC_EC_AnonymousCallRejection	116
CC_EC_BaercapNotAvail	3
CC_EC_BearcapNotAuthorized	2
CC_EC_BearcapNotImp	4
CC_EC_BlacklistBNumberMatched	137
CC_EC_BlacklistCliLengthInvalid	133
CC_EC_BlacklistCliMatched	134
CC_EC_BlacklistCpcRestricted	135
CC_EC_BlacklistNoCli	132
CC_EC_BlacklistNoaRestricted	136
CC_EC_Busy	57
CC_EC_CallAwardedDeliveredEstCh	5
CC_EC_CallIdHasBeenCleared	6
CC_EC_CallIdInUse	7
CC_EC_CallRejectCallGapping	102
CC_EC_CallRejected	8
CC_EC_CallTerminated	61
CC_EC_CallTypeIncompatible	95
CC_EC_CallingDroppedWhileOnHold	98
CC_EC_CallingPartyOffHold	97
CC_EC_ChIdNotExist	9
CC_EC_ChTypeNotImp	10
CC_EC_ChUnacceptable	11
CC_EC_ChannelOutOfService	58
CC_EC_Congestion	60
CC_EC_CotFailure	141
CC_EC_CugAccessBarred	108
CC_EC_DEFAULT	32
CC_EC_DestOutOfOrder	12
CC_EC_DteControlledNotReady	59
CC_EC_DteUncontrolledNotReady	79
CC_EC_ElemTypeNotImp	13
CC_EC_ExcessiveDigCallProceeding	123
CC_EC_FacilityNotRegistered	62
CC_EC_FacilityRejected	14
CC_EC_FlowControlledCongestion	110
CC_EC_GroupRestrictions	96

CC_EC_IncomingCallsBarred	63
CC_EC_IncomingCallsBarredInCug	81
CC_EC_IncompatibleDest	15
CC_EC_InconOutgoingAccAndSubClass	126
CC_EC_InterceptedSubscriber	53
CC_EC_InterworkUnspec	16
CC_EC_InvalidCallRef	143
CC_EC_InvalidCallReferenceValue	17
CC_EC_InvalidElemContents	18
CC_EC_InvalidMsgUnspec	19
CC_EC_InvalidNumberFormat	20
CC_EC_InvalidTns	21
CC_EC_MandatoryElementMissing	22
CC_EC_MesgWithUnrecElemDiscarded	128
CC_EC_MessageNotUnderstood	65
CC_EC_MisdialledTkPrefix	84
CC_EC_MisroutedCallPortedNumber	142
CC_EC_MsgInWrongState	23
CC_EC_MsgTypeNotImp	24
CC_EC_MsgTypeNotImpOrWrongState	25
CC_EC_NetworkAddressExtensionError	66
CC_EC_NetworkOutOfOrder	26
CC_EC_NetworkTermination	67
CC_EC_NewDestination	99
CC_EC_NoAnswerAlertedUser	28
CC_EC_NoCallSuspended	27
CC_EC_NoCircuitAvailable	29
CC_EC_NoRouteToDest	33
CC_EC_NoRouteToTns	34
CC_EC_NoUserResponding	35
CC_EC_NonExistentCug	127
CC_EC_NonSelectedUserClearing	30
CC_EC_NormalClearing	31
CC_EC_NormalUnspecified	32
CC_EC_NumberChanged	36
CC_EC_NumberUnobtainable	68
CC_EC_OnlyRestrictDigInfoBearer	37
CC_EC_OperatorPriorityAccess	107
CC_EC_OutOfCatchmentArea	111
CC_EC_OutgoingCallsBarred	100
CC_EC_OutgoingCallsBarredInCug	125
CC_EC_ParamUnrecPassed	85
CC_EC_PermanentFrameModeOos	130
CC_EC_PermanentFrameModeOperational	131
CC_EC_PermanentIcb	113
CC_EC_PortedNumber	139
CC_EC_PrecedenceBlocked	94
CC_EC_Preemption	87
CC_EC_PreemptionCctRes	129
CC_EC_PreemptionCctUnavailable	88
CC_EC_Prefix0DialledInError	120
CC_EC_Prefix1DialledInError	121
CC_EC_Prefix1NotDialled	122
CC_EC_PriorityForcedRelease	69
CC_EC_Proprietary	86
CC_EC_ProtErrThresholdExceeded	124
CC_EC_ProtocolErrorUnspec	38
CC_EC_QualityUnavail	39
CC_EC_RecoveryOnTimerExpiry	40
CC_EC_RedirectionToNewDest	140
CC_EC_Reject	70
CC_EC_RejectedDivertedCall	103
CC_EC_RemoteProcError	105
CC_EC_RepeatAttempt	118

CC_EC_ReqCircuitUnavail	41
CC_EC_ReqFacilityNotImp	42
CC_EC_ReqFacilityNotSubscr	43
CC_EC_ResourcesUnavailUnspec	44
CC_EC_ResponseToStatusEnquiry	45
CC_EC_RouteOutOfService	71
CC_EC_RoutingError	93
CC_EC_SelectiveCallBarring	104
CC_EC_ServiceIncompatible	64
CC_EC_ServiceOrOptionNotAvail	47
CC_EC_ServiceOrOptionNotImpUnspec	46
CC_EC_ServiceTemporarilyUnavailable	77
CC_EC_ServiceUnavailable	78
CC_EC_SignalNotUnderstood	73
CC_EC_SignalNotValid	74
CC_EC_SignallingSystemIncompatible	76
CC_EC_SpecialInformationTone	82
CC_EC_SubAbscent	91
CC_EC_SubControlledIcb	101
CC_EC_SubNotFoundDle	115
CC_EC_SubscriberCallTerminate	109
CC_EC_SubscriberIncompatible	72
CC_EC_SubscriberMoved	114
CC_EC_SubscriberOutOfService	75
CC_EC_SuspendExistButNotThisId	48
CC_EC_SwitchingEquipCongestion	49
CC_EC_TemporaryFailure	50
CC_EC_TemporaryOos	106
CC_EC_TerminalCongestion	117
CC_EC_Transferred	80
CC_EC_TranslationOos	112
CC_EC_UnallocatedDestNumber	89
CC_EC_UnallocatedNumber	51
CC_EC_UndefinedBg	92
CC_EC_Unknown	144
CC_EC_UnrecElemPassedOn	90
CC_EC_UserBusy	52
CC_EC_UserNotMemberOfCug	83
CC_EC_VacentCode	119
CC_EC_WhitelistCliNotMatched	138

E-ISUP Cause Value-to-Name Lookup

This appendix lists the Cisco E-ISUP cause values and their associated names.

Value	Name
1	CC_EC_AccessInfoDiscarded
2	CC_EC_BearcapNotAuthorized
3	CC_EC_BearcapNotAvail
4	CC_EC_BearcapNotImp
5	CC_EC_CallAwardedDeliveredEstCh
6	CC_EC_CallIdHasBeenCleared
7	CC_EC_CallIdInUse
8	CC_EC_CallRejected
9	CC_EC_ChIdNotExist
10	CC_EC_ChTypeNotImp
11	CC_EC_ChUnacceptable
12	CC_EC_DestOutOfOrder
13	CC_EC_ElemTypeNotImp
14	CC_EC_FacilityRejected
15	CC_EC_IncompatibleDest
16	CC_EC_InterworkUnspec
17	CC_EC_InvalidCallReferenceValue
18	CC_EC_InvalidElemContents
19	CC_EC_InvalidMsgUnspec
20	CC_EC_InvalidNumberFormat
21	CC_EC_InvalidTns
22	CC_EC_MandatoryElementMissing
23	CC_EC_MsgInWrongState
24	CC_EC_MsgTypeNotImp
25	CC_EC_MsgTypeNotImpOrWrongState
26	CC_EC_NetworkOutOfOrder
27	CC_EC_NoCallSuspended
28	CC_EC_NoAnswerAlertedUser
29	CC_EC_NoCircuitAvailable
30	CC_EC_NonSelectedUserClearing
31	CC_EC_NormalClearing
32	CC_EC_DEFAULT
32	CC_EC_NormalUnspecified
33	CC_EC_NoRouteToDest
34	CC_EC_NoRouteToTns
35	CC_EC_NoUserResponding
36	CC_EC_NumberChanged
37	CC_EC_OnlyRestrictDigInfoBearer
38	CC_EC_ProtocolErrorUnspec
39	CC_EC_QualityUnavail
40	CC_EC_RecoveryOnTimerExpiry
41	CC_EC_ReqCircuitUnavail

```
 42  CC_EC_ReqFacilityNotImp
 43  CC_EC_ReqFacilityNotSubscr
 44  CC_EC_ResourcesUnavailUnspec
 45  CC_EC_ResponseToStatusEnquiry
 46  CC_EC_ServiceOrOptionNotImpUnspec
 47  CC_EC_ServiceOrOptionNotAvail
 48  CC_EC_SuspendExistButNotThisId
 49  CC_EC_SwitchingEquipCongestion
 50  CC_EC_TemporaryFailure
 51  CC_EC_UnallocatedNumber
 52  CC_EC_UserBusy
 53  CC_EC_InterceptedSubscriber
 54  CC_EC_AccessBarred
 55  CC_EC_Acknowledgement
 56  CC_EC_AddressIncomplete
 57  CC_EC_Busy
 58  CC_EC_ChannelOutOfService
 59  CC_EC_DteControlledNotReady
 60  CC_EC_Congestion
 61  CC_EC_CallTerminated
 62  CC_EC_FacilityNotRegistered
 63  CC_EC_IncomingCallsBarred
 64  CC_EC_ServiceIncompatible
 65  CC_EC_MessageNotUnderstood
 66  CC_EC_NetworkAddressExtensionError
 67  CC_EC_NetworkTermination
 68  CC_EC_NumberUnobtainable
 69  CC_EC_PriorityForcedRelease
 70  CC_EC_Reject
 71  CC_EC_RouteOutOfService
 72  CC_EC_SubscriberIncompatible
 73  CC_EC_SignalNotUnderstood
 74  CC_EC_SignalNotValid
 75  CC_EC_SubscriberOutOfService
 76  CC_EC_SignallingSystemIncompatible
 77  CC_EC_ServiceTemporarilyUnavailable
 78  CC_EC_ServiceUnavailable
 79  CC_EC_DteUncontrolledNotReady
 80  CC_EC_Transferred
 81  CC_EC_IncomingCallsBarredInCug
 82  CC_EC_SpecialInformationTone
 83  CC_EC_UserNotMemberOfCug
 84  CC_EC_MisdialledTkPrefix
 85  CC_EC_ParamUnrecPassed
 86  CC_EC_Proprietary
 87  CC_EC_Preemption
 88  CC_EC_PreemptionCctUnavailable
 89  CC_EC_UnallocatedDestNumber
 90  CC_EC_UnrecElemPassedOn
 91  CC_EC_SubAbscent
 92  CC_EC_UndefinedBg
 93  CC_EC_RoutingError
 94  CC_EC_PrecedenceBlocked
 95  CC_EC_CallTypeIncompatible
 96  CC_EC_GroupRestrictions
 97  CC_EC_CallingPartyOffHold
 98  CC_EC_CallingDroppedWhileOnHold
 99  CC_EC_NewDestination
100  CC_EC_OutgoingCallsBarred
101  CC_EC_SubControlledIcb
102  CC_EC_CallRejectCallGapping
103  CC_EC_RejectedDivertedCall
104  CC_EC_SelectiveCallBarring
105  CC_EC_RemoteProcError
```

```
106  CC_EC_TemporaryOos
107  CC_EC_OperatorPriorityAccess
108  CC_EC_CugAccessBarred
109  CC_EC_SubscriberCallTerminate
110  CC_EC_FlowControlledCongestion
111  CC_EC_OutOfCatchmentArea
112  CC_EC_TranslationOos
113  CC_EC_PermanentIcb
114  CC_EC_SubscriberMoved
115  CC_EC_SubNotFoundDle
116  CC_EC_AnonymousCallRejection
117  CC_EC_TerminalCongestion
118  CC_EC_RepeatAttempt
119  CC_EC_VacentCode
120  CC_EC_Prefix0DialledInError
121  CC_EC_Prefix1DialledInError
122  CC_EC_Prefix1NotDialled
123  CC_EC_ExcessiveDigCallProceeding
124  CC_EC_ProtErrThresholdExceeded
125  CC_EC_OutgoingCallsBarredInCug
126  CC_EC_InconOutgoingAccAndSubClass
127  CC_EC_NonExistentCug
128  CC_EC_MesgWithUnrecElemDiscarded
129  CC_EC_PreemptionCctRes
130  CC_EC_PermanentFrameModeOos
131  CC_EC_PermanentFrameModeOperational
132  CC_EC_BlacklistNoCli
133  CC_EC_BlacklistCliLengthInvalid
134  CC_EC_BlacklistCliMatched
135  CC_EC_BlacklistCpcRestricted
136  CC_EC_BlacklistNoaRestricted
137  CC_EC_BlacklistBNumberMatched
138  CC_EC_WhitelistCliNotMatched
139  CC_EC_PortedNumber
140  CC_EC_RedirectionToNewDest
141  CC_EC_CotFailure
142  CC_EC_MisroutedCallPortedNumber
143  CC_EC_InvalidCallRef
144  CC_EC_Unknown
```

H.323 Name-to-Cause Value Lookup

This appendix lists the H.323 names and their associated cause values.

Name	Value
CC_HC_ASuspendedCallExistsThisCallIdDoesNot	83
CC_HC_AccessInformationDiscarded	43
CC_HC_BearerCapabilityNotAuthorized	57
CC_HC_BearerCapabilityNotImplemented	65
CC_HC_BearerCapabilityNotPresentlyAvailable	58
CC_HC_CallAwardedEstablishedChannel	7
CC_HC_CallHavingTheRequestedCallIdHasBeenCleared	86
CC_HC_CallIdentityInUse	84
CC_HC_CallRejected	21
CC_HC_ChannelTypeNotImplemented	66
CC_HC_ChannelUnacceptable	6
CC_HC_DEFAULT	31
CC_HC_DestinationOutOfOrder	27
CC_HC_ExchangeRouteError	25
CC_HC_FacilityRejected	29
CC_HC_IdentifiedChannelDoesnotExist	82
CC_HC_IncomingCallsBaredWithinCUG	55
CC_HC_IncompatibleDestination	88
CC_HC_InconsistencyAcessInfoSubscriberClass	62
CC_HC_InformationElementParameterNonExistantOrNotImplemented	99
CC_HC_InvalidCallreferenceValue	81
CC_HC_InvalidInformationElementContents	100
CC_HC_InvalidMessage	95
CC_HC_InvalidNumberFormat	28
CC_HC_InvalidTransitNetworkSelection	91
CC_HC_MandatoryInformationElementIsMissing	96
CC_HC_MessageNotCompatibleWithCallState	101
CC_HC_MessageTypeNonExistantOrNotImplemented	97
CC_HC_MessageTypeNotCompatibleWithStateOrNonExistantOrNotImplemented	98
CC_HC_MisdialedTrunkPrefix	5
CC_HC_NetworkOutOfOrder	38
CC_HC_NoAnswerFromAlertedUser	19
CC_HC_NoCallSuspended	85
CC_HC_NoCircuitChannelAvailable	34
CC_HC_NoRouteToDestination	3
CC_HC_NoRouteToSpecifiedTransitNetwork	2
CC_HC_NoUserresponding	18
CC_HC_NonExistantCUG	90
CC_HC_NonSelectedUserClearing	26
CC_HC_NormalCallClearing	16
CC_HC_NormalUnspecified	31

CC_HC_NumberChanged	22
CC_HC_OnlyRestrictedDigitalBearerInfoCapability	70
CC_HC_OutgoingCallsBaredWithinCUG	53
CC_HC_ParameterNonExistantOrNotImplementedPassedOn	103
CC_HC_PermanentFrameModeConnectionOperational	40
CC_HC_PermanentFrameModeConnectionOutOfService	39
CC_HC_PrecedenceCallBlocked	46
CC_HC_Preemption	8
CC_HC_PreemptionCircuitReservedForReuse	9
CC_HC_QualityOfServiceNotAvailable	49
CC_HC_RecoveryOnTimerExpiry	102
CC_HC_RedirectionToNewDestination	23
CC_HC_RequestedCircuitChannelNotAvailable	44
CC_HC_RequestedFacilityNotImplemented	69
CC_HC_RequestedFacilityNotSubscribed	50
CC_HC_ResourceUnavailable	47
CC_HC_ResponceToStatusEnquiry	30
CC_HC_SendSpecialInformationTone	4
CC_HC_ServiceOrOptionNotImplemented	79
CC_HC_ServiceOrOptionUnavailable	63
CC_HC_SubscriberAbsent	20
CC_HC_SwitchingEquipmentCongestion	42
CC_HC_TemporaryFailure	41
CC_HC_UnallocatedNumber	1
CC_HC_UserBusy	17
CC_HC_UserNotMemberOfCUG	87

H.323 Cause Value-to-Name Lookup

This appendix lists the H.323 cause values and their associated names.

Value	Name
1	CC_HC_UnallocatedNumber
2	CC_HC_NoRouteToSpecifiedTransitNetwork
3	CC_HC_NoRouteToDestination
4	CC_HC_SendSpecialInformationTone
5	CC_HC_MisdialedTrunkPrefix
6	CC_HC_ChannelUnacceptable
7	CC_HC_CallAwardedEstablishedChannel
8	CC_HC_Preemption
9	CC_HC_PreemptionCircuitReservedForReuse
16	CC_HC_NormalCallClearing
17	CC_HC_UserBusy
18	CC_HC_NoUserresponding
19	CC_HC_NoAnswerFromAlertedUser
20	CC_HC_SubscriberAbsent
21	CC_HC_CallRejected
22	CC_HC_NumberChanged
23	CC_HC_RedirectionToNewDestination
25	CC_HC_ExchangeRouteError
26	CC_HC_NonSelectedUserClearing
27	CC_HC_DestinationOutOfOrder
28	CC_HC_InvalidNumberFormat
29	CC_HC_FacilityRejected
30	CC_HC_ResponceToStatusEnquiry
31	CC_HC_DEFAULT
31	CC_HC_NormalUnspecified
34	CC_HC_NoCircuitChannelAvailable
38	CC_HC_NetworkOutOfOrder
39	CC_HC_PermanentFrameModeConnectionOutOfService
40	CC_HC_PermanentFrameModeConnectionOperational
41	CC_HC_TemporaryFailure
42	CC_HC_SwitchingEquipmentCongestion
43	CC_HC_AccessInformationDiscarded
44	CC_HC_RequestedCircuitChannelNotAvailable
46	CC_HC_PrecedenceCallBlocked
47	CC_HC_ResourceUnavailable
49	CC_HC_QualityOfServiceNotAvailable
50	CC_HC_RequestedFacilityNotSubscribed
53	CC_HC_OutgoingCallsBaredWithinCUG
55	CC_HC_IncomingCallsBaredWithinCUG
57	CC_HC_BearerCapabilityNotAuthorized
58	CC_HC_BearerCapabilityNotPresentlyAvailable
62	CC_HC_InconsistencyAcessInfoSubscriberClass

```
 63  CC_HC_ServiceOrOptionUnavailable
 65  CC_HC_BearerCapabilityNotImplemented
 66  CC_HC_ChannelTypeNotImplemented
 69  CC_HC_RequestedFacilityNotImplemented
 70  CC_HC_OnlyRestrictedDigitalBearerInfoCapability
 79  CC_HC_ServiceOrOptionNotImplemented
 81  CC_HC_InvalidCallreferenceValue
 82  CC_HC_IdentifiedChannelDoesnotExist
 83  CC_HC_ASuspendedCallExistsThisCallIdDoesNot
 84  CC_HC_CallIdentityInUse
 85  CC_HC_NoCallSuspended
 86  CC_HC_CallHavingTheRequestedCallIdHasBeenCleared
 87  CC_HC_UserNotMemberOfCUG
 88  CC_HC_IncompatibleDestination
 90  CC_HC_NonExistantCUG
 91  CC_HC_InvalidTransitNetworkSelection
 95  CC_HC_InvalidMessage
 96  CC_HC_MandatoryInformationElementIsMissing
 97  CC_HC_MessageTypeNonExistantOrNotImplemented
 98  CC_HC_MessageTypeNotCompatibleWithStateOrNonExistantOrNotImplemented
 99  CC_HC_InformationElementParameterNonExistantOrNotImplemented
100  CC_HC_InvalidInformationElementContents
101  CC_HC_MessageNotCompatibleWithCallState
102  CC_HC_RecoveryOnTimerExpiry
103  CC_HC_ParameterNonExistantOrNotImplementedPassedOn
```

A

B

C

D

E

G

H

I

L

M

N

O

P

Q

R

S

T

U

V